

LinuxQMISDK  
SLQS04.00.01

Generated by Doxygen 1.8.6

Wed Aug 24 2016 10:19:30



# 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>27</b>
5.1	File List . . . . .	27
<b>6</b>	<b>Module Documentation</b>	<b>31</b>
6.1	Device Connectivity Service (DCS) . . . . .	31
6.1.1	Detailed Description . . . . .	31
6.2	Wireless Data Service (WDS) . . . . .	32
6.2.1	Detailed Description . . . . .	32
6.3	Device Management Service (DMS) . . . . .	33
6.3.1	Detailed Description . . . . .	33
6.4	Network Access Service (NAS) . . . . .	34
6.4.1	Detailed Description . . . . .	34
6.5	CallBack registration (CBK) . . . . .	35
6.5.1	Detailed Description . . . . .	35
6.6	Short Message Service (SMS) . . . . .	36
6.6.1	Detailed Description . . . . .	36

6.7	Position Determination Service (PDS)	37
6.7.1	Detailed Description	37
6.8	Card Application Toolkit (CAT)	38
6.8.1	Detailed Description	38
6.9	Remote Management Service (RMS)	39
6.9.1	Detailed Description	39
6.10	Firmware Management Service (FMS)	40
6.10.1	Detailed Description	40
6.11	Open Mobile Alliance Service (OMA)	41
6.11.1	Detailed Description	41
6.12	Specific Absorption Rate (SAR)	42
6.12.1	Detailed Description	42
6.13	SWI Open Mobile Alliance Service (SWIOMA)	43
6.13.1	Detailed Description	43
6.14	Voice Service (VOICE)	44
6.14.1	Detailed Description	44
6.15	Non-service specific APIs (SWI)	45
6.15.1	Detailed Description	45
6.16	User Identity Module Service (UIM)	46
6.16.1	Detailed Description	46
6.17	Audio Service (AUDIO)	47
6.17.1	Detailed Description	47
6.18	Quality of Service (QOS)	48
6.18.1	Detailed Description	48
6.19	IMS Service (IMS)	49
6.19.1	Detailed Description	49
6.20	SWI Audio Service(SWIAUDIO)	50
6.20.1	Detailed Description	50
6.21	Location Service(LOC)	51
6.21.1	Detailed Description	51
6.22	Thermal Mitigation Device(TMD)	52
6.22.1	Detailed Description	52
<b>7</b>	<b>Namespace Documentation</b>	<b>53</b>
7.1	Tables Namespace Reference	53
7.1.1	Detailed Description	53
<b>8</b>	<b>Data Structure Documentation</b>	<b>55</b>
8.1	_getIndicationRegResp Struct Reference	55
8.1.1	Detailed Description	55
8.1.2	Field Documentation	56

8.1.2.1	<a href="#">pRegCallStatInfoEvt</a>	56
8.1.2.2	<a href="#">pRegTransLayerInfoEvt</a>	56
8.1.2.3	<a href="#">pRegTransNWRegInfoEvt</a>	56
8.2	<a href="#">_GetProfileSettingIn Struct Reference</a>	56
8.2.1	<a href="#">Detailed Description</a>	56
8.2.2	<a href="#">Field Documentation</a>	56
8.2.2.1	<a href="#">ProfileID</a>	56
8.2.2.2	<a href="#">ProfileType</a>	56
8.3	<a href="#">_GetProfileSettingOut Struct Reference</a>	56
8.3.1	<a href="#">Detailed Description</a>	56
8.3.2	<a href="#">Field Documentation</a>	57
8.3.2.1	<a href="#">curProfile</a>	57
8.3.2.2	<a href="#">pExtErrCode</a>	57
8.4	<a href="#">_getResetInfoNotification Struct Reference</a>	57
8.4.1	<a href="#">Detailed Description</a>	57
8.4.2	<a href="#">Field Documentation</a>	58
8.4.2.1	<a href="#">source</a>	58
8.4.2.2	<a href="#">type</a>	58
8.5	<a href="#">_getTransLayerInfoResp Struct Reference</a>	58
8.5.1	<a href="#">Detailed Description</a>	58
8.5.2	<a href="#">Field Documentation</a>	58
8.5.2.1	<a href="#">pRegInd</a>	58
8.5.2.2	<a href="#">pTransLayerInfo</a>	58
8.6	<a href="#">_getTransNWRegInfoResp Struct Reference</a>	58
8.6.1	<a href="#">Detailed Description</a>	58
8.6.2	<a href="#">Field Documentation</a>	59
8.6.2.1	<a href="#">pRegStatus</a>	59
8.7	<a href="#">_MitigationDevInfo Struct Reference</a>	59
8.7.1	<a href="#">Detailed Description</a>	59
8.7.2	<a href="#">Field Documentation</a>	59
8.7.2.1	<a href="#">deviceId</a>	59
8.7.2.2	<a href="#">deviceldLen</a>	59
8.8	<a href="#">_modemTempNotification Struct Reference</a>	59
8.8.1	<a href="#">Detailed Description</a>	60
8.8.2	<a href="#">Field Documentation</a>	60
8.8.2.1	<a href="#">ModemTemperature</a>	60
8.8.2.2	<a href="#">ModemTempState</a>	60
8.9	<a href="#">_packetSrvStatus Struct Reference</a>	60
8.9.1	<a href="#">Detailed Description</a>	60
8.9.2	<a href="#">Field Documentation</a>	62

8.9.2.1	bearerID	62
8.9.2.2	connStatus	62
8.9.2.3	ipFamily	62
8.9.2.4	pQmiInterfaceInfo	62
8.9.2.5	reconfigReqd	62
8.9.2.6	sessionEndReason	62
8.9.2.7	techName	62
8.9.2.8	verboseSessnEndReason	62
8.9.2.9	verboseSessnEndReasonType	62
8.10	_qaQmi3GPP2BroadcastCfgInfo Struct Reference	62
8.10.1	Detailed Description	62
8.10.2	Field Documentation	63
8.10.2.1	activated_ind	63
8.10.2.2	CDMABroadcastConfig	63
8.10.2.3	num_instances	63
8.11	_qaQmi3GPPBroadcastCfgInfo Struct Reference	63
8.11.1	Detailed Description	63
8.11.2	Field Documentation	63
8.11.2.1	activated_ind	63
8.11.2.2	broadcastConfig	63
8.11.2.3	num_instances	63
8.12	_setIndicationRegReq Struct Reference	63
8.12.1	Detailed Description	64
8.12.2	Field Documentation	64
8.12.2.1	pRegCallStatInfoEvt	64
8.12.2.2	pRegTransLayerInfoEvt	64
8.12.2.3	pRegTransNWRegInfoEvt	64
8.13	_slqs3GPPConfigItem Struct Reference	64
8.13.1	Detailed Description	65
8.13.2	Field Documentation	66
8.13.2.1	LTEAttachProfileListLen	66
8.13.2.2	p3gppRelease	66
8.13.2.3	pDefaultPDNEnabled	66
8.13.2.4	pLTEAttachProfile	66
8.13.2.5	pLTEAttachProfileList	66
8.13.2.6	pProfileList	66
8.14	_SlqsNas3GppNetworkRAT_ Struct Reference	66
8.14.1	Detailed Description	66
8.14.2	Field Documentation	66
8.14.2.1	MCC	66

8.14.2.2	MNC	67
8.14.2.3	RAT	67
8.15	_slqsNetworkScanInfo Struct Reference	67
8.15.1	Detailed Description	67
8.15.2	Field Documentation	68
8.15.2.1	pNetworkInfo	68
8.15.2.2	pNetworkInfoInstances	68
8.15.2.3	pPCSDigitInfo	68
8.15.2.4	pPCSDigitInstances	68
8.15.2.5	pRATInfo	68
8.15.2.6	pRATInstances	68
8.15.2.7	pScanResult	68
8.16	_SLQSOMADMSessionInfo Struct Reference	68
8.16.1	Detailed Description	68
8.16.2	Field Documentation	70
8.16.2.1	pDate	70
8.16.2.2	pDateLength	70
8.16.2.3	pPkgDescLength	70
8.16.2.4	pPkgDescription	70
8.16.2.5	pPkgName	70
8.16.2.6	pPkgNameLength	70
8.16.2.7	pRetryCount	70
8.16.2.8	pSessionState	70
8.16.2.9	pSessionType	70
8.16.2.10	pSeverity	70
8.16.2.11	pSource	70
8.16.2.12	pSourceLength	70
8.16.2.13	pStatus	70
8.16.2.14	pTime	70
8.16.2.15	pTimeLength	71
8.16.2.16	pUpdateCompleteStatus	71
8.17	_SLQSOMADMSettings Struct Reference	71
8.17.1	Detailed Description	71
8.17.2	Field Documentation	72
8.17.2.1	pAutosdm	72
8.17.2.2	pFOTAdownload	72
8.17.2.3	pFOTAUpdate	72
8.17.2.4	pFwAutoCheck	72
8.17.2.5	pOMADMEEnabled	72
8.18	_SLQSOMADMSettingsReqParams Struct Reference	72

8.18.1 Detailed Description . . . . .	72
8.18.2 Field Documentation . . . . .	73
8.18.2.1 FOTAdownload . . . . .	73
8.18.2.2 FOTAUpdate . . . . .	73
8.18.2.3 pAutosdm . . . . .	73
8.19 _SLQSOMADMSettingsReqParams3 Struct Reference . . . . .	73
8.19.1 Detailed Description . . . . .	73
8.19.2 Field Documentation . . . . .	74
8.19.2.1 FOTAdownload . . . . .	74
8.19.2.2 FOTAUpdate . . . . .	74
8.19.2.3 pAutosdm . . . . .	74
8.19.2.4 pFwAutoCheck . . . . .	74
8.20 _SLQSSwiGetHostDevInfoParams Struct Reference . . . . .	74
8.20.1 Detailed Description . . . . .	74
8.20.2 Field Documentation . . . . .	75
8.20.2.1 bManSize . . . . .	75
8.20.2.2 bModelSize . . . . .	75
8.20.2.3 bPlasmaIDSize . . . . .	75
8.20.2.4 bSWVerSize . . . . .	75
8.20.2.5 pManString . . . . .	75
8.20.2.6 pModelString . . . . .	75
8.20.2.7 pPlasmaIDString . . . . .	75
8.20.2.8 pSWVerString . . . . .	75
8.21 _SLQSSwiGetOSInfoParams Struct Reference . . . . .	75
8.21.1 Detailed Description . . . . .	76
8.21.2 Field Documentation . . . . .	76
8.21.2.1 bNameSize . . . . .	76
8.21.2.2 bVersionSize . . . . .	76
8.21.2.3 pNameString . . . . .	76
8.21.2.4 pVersionString . . . . .	76
8.22 _SLQSSwiGetSerialNoExtParams Struct Reference . . . . .	76
8.22.1 Detailed Description . . . . .	76
8.22.2 Field Documentation . . . . .	77
8.22.2.1 meidLength . . . . .	77
8.22.2.2 pMeidString . . . . .	77
8.23 _SLQSSwiSetHostDevInfoParams Struct Reference . . . . .	77
8.23.1 Detailed Description . . . . .	77
8.23.2 Field Documentation . . . . .	78
8.23.2.1 bManSize . . . . .	78
8.23.2.2 bModelSize . . . . .	78

8.23.2.3	bPlasmaIDSize . . . . .	78
8.23.2.4	bSWVerSize . . . . .	78
8.23.2.5	pManString . . . . .	78
8.23.2.6	pModelString . . . . .	78
8.23.2.7	pPlasmaIDString . . . . .	78
8.23.2.8	pSWVerString . . . . .	78
8.24	_SLQSSwiSetOSInfoParams Struct Reference . . . . .	78
8.24.1	Detailed Description . . . . .	78
8.24.2	Field Documentation . . . . .	79
8.24.2.1	bNameSize . . . . .	79
8.24.2.2	bVersionSize . . . . .	79
8.24.2.3	pNameString . . . . .	79
8.24.2.4	pVersionString . . . . .	79
8.25	_sysSelectPrefInfo Struct Reference . . . . .	79
8.25.1	Detailed Description . . . . .	79
8.25.2	Field Documentation . . . . .	84
8.25.2.1	pBandPref . . . . .	84
8.25.2.2	pEmerMode . . . . .	84
8.25.2.3	pGWAcqOrderPref . . . . .	84
8.25.2.4	pLTEBandPref . . . . .	84
8.25.2.5	pModePref . . . . .	84
8.25.2.6	pNetSelPref . . . . .	84
8.25.2.7	pPRLPref . . . . .	84
8.25.2.8	pRoamPref . . . . .	84
8.25.2.9	pSrvDomainPref . . . . .	84
8.26	_sysSelectPrefParams Struct Reference . . . . .	85
8.26.1	Detailed Description . . . . .	85
8.26.2	Field Documentation . . . . .	90
8.26.2.1	pAcqOrderPref . . . . .	90
8.26.2.2	pBandPref . . . . .	90
8.26.2.3	pChgDuration . . . . .	90
8.26.2.4	pCSGID . . . . .	90
8.26.2.5	pEmerMode . . . . .	90
8.26.2.6	pGWAcqOrderPref . . . . .	90
8.26.2.7	pLTEBandPref . . . . .	90
8.26.2.8	pMNCIncPCSDigStat . . . . .	90
8.26.2.9	pModePref . . . . .	90
8.26.2.10	pNetSelPref . . . . .	90
8.26.2.11	pPRLPref . . . . .	90
8.26.2.12	pRAT . . . . .	90

8.26.2.13	pRoamPref	91
8.26.2.14	pSrvDomainPref	91
8.26.2.15	pSrvRegRestriction	91
8.26.2.16	pTdsdmaBandPref	91
8.27	_transLayerinfo Struct Reference	91
8.27.1	Detailed Description	91
8.27.2	Field Documentation	91
8.27.2.1	TransCap	91
8.27.2.2	TransType	91
8.28	_transLayerInfoNotification Struct Reference	91
8.28.1	Detailed Description	91
8.28.2	Field Documentation	92
8.28.2.1	pTransLayerInfo	92
8.28.2.2	regInd	92
8.29	_transNWRegInfoNotification Struct Reference	92
8.29.1	Detailed Description	92
8.29.2	Field Documentation	92
8.29.2.1	NWRegStat	92
8.30	accelAcceptReady_s Struct Reference	93
8.30.1	Detailed Description	93
8.30.2	Field Documentation	93
8.30.2.1	batchPerSec	93
8.30.2.2	injectEnable	93
8.30.2.3	samplesPerBatch	93
8.31	accelTempAcceptReady_s Struct Reference	93
8.31.1	Detailed Description	93
8.31.2	Field Documentation	94
8.31.2.1	batchPerSec	94
8.31.2.2	injectEnable	94
8.31.2.3	samplesPerBatch	94
8.32	acqOrderPref Struct Reference	94
8.32.1	Detailed Description	94
8.32.2	Field Documentation	95
8.32.2.1	acqOrdeLen	95
8.32.2.2	pAcqOrder	95
8.33	ActPilotPNElement Struct Reference	95
8.33.1	Detailed Description	95
8.33.2	Field Documentation	95
8.33.2.1	ActSetPilotPN	95
8.33.2.2	ActSetPilotPNStrength	95

8.34 AddCDMASysInfo Struct Reference . . . . .	95
8.34.1 Detailed Description . . . . .	95
8.34.2 Field Documentation . . . . .	96
8.34.2.1 geoSysIdx . . . . .	96
8.34.2.2 regPrd . . . . .	96
8.35 AddSysInfo Struct Reference . . . . .	96
8.35.1 Detailed Description . . . . .	96
8.35.2 Field Documentation . . . . .	96
8.35.2.1 cellBroadcastCap . . . . .	96
8.35.2.2 geoSysIdx . . . . .	96
8.36 airTimer Struct Reference . . . . .	96
8.36.1 Detailed Description . . . . .	96
8.36.2 Field Documentation . . . . .	97
8.36.2.1 airTimerValue . . . . .	97
8.36.2.2 namID . . . . .	97
8.37 allCallsAlphaIDInfo Struct Reference . . . . .	97
8.37.1 Detailed Description . . . . .	97
8.37.2 Field Documentation . . . . .	97
8.37.2.1 AlphaIDInfo . . . . .	97
8.37.2.2 callID . . . . .	97
8.38 allCallsDiagInfo Struct Reference . . . . .	97
8.38.1 Detailed Description . . . . .	98
8.38.2 Field Documentation . . . . .	98
8.38.2.1 callID . . . . .	98
8.38.2.2 DiagInfo . . . . .	98
8.39 allCallsUUSInfo Struct Reference . . . . .	98
8.39.1 Detailed Description . . . . .	98
8.39.2 Field Documentation . . . . .	98
8.39.2.1 callID . . . . .	98
8.39.2.2 uusInfo . . . . .	98
8.40 alphaIDInfo Struct Reference . . . . .	98
8.40.1 Detailed Description . . . . .	99
8.40.2 Field Documentation . . . . .	99
8.40.2.1 alphaDcs . . . . .	99
8.40.2.2 alphaLen . . . . .	99
8.40.2.3 alphaText . . . . .	99
8.41 altitudeSrcInfo Struct Reference . . . . .	99
8.41.1 Detailed Description . . . . .	99
8.41.2 Field Documentation . . . . .	100
8.41.2.1 coverage . . . . .	100

8.41.2.2	linkage	100
8.41.2.3	source	100
8.42	appStats Struct Reference	100
8.42.1	Detailed Description	101
8.42.2	Field Documentation	103
8.42.2.1	aidLength	103
8.42.2.2	aidVal	103
8.42.2.3	appState	103
8.42.2.4	appType	103
8.42.2.5	persoFeature	103
8.42.2.6	persoRetries	103
8.42.2.7	persoState	103
8.42.2.8	persoUnblockRetries	103
8.42.2.9	pin1Retries	103
8.42.2.10	pin1State	103
8.42.2.11	pin2Retries	103
8.42.2.12	pin2State	103
8.42.2.13	puk1Retries	103
8.42.2.14	puk2Retries	103
8.42.2.15	univPin	103
8.43	appStatus Struct Reference	103
8.43.1	Detailed Description	104
8.43.2	Field Documentation	106
8.43.2.1	aidLength	106
8.43.2.2	aidVal	106
8.43.2.3	appState	106
8.43.2.4	appType	106
8.43.2.5	persoFeature	106
8.43.2.6	persoRetries	106
8.43.2.7	persoState	106
8.43.2.8	persoUnblockRetries	106
8.43.2.9	pin1Retries	106
8.43.2.10	pin1State	106
8.43.2.11	pin2Retries	106
8.43.2.12	pin2State	106
8.43.2.13	puk1Retries	106
8.43.2.14	puk2Retries	106
8.43.2.15	univPin	106
8.44	arrAlertingPattern Struct Reference	106
8.44.1	Detailed Description	107

8.44.2	Field Documentation	107
8.44.2.1	alertingPattern	107
8.44.2.2	callID	107
8.44.2.3	numInstances	107
8.45	arrAlertingType Struct Reference	107
8.45.1	Detailed Description	107
8.45.2	Field Documentation	108
8.45.2.1	AlertingType	108
8.45.2.2	callID	108
8.45.2.3	numInstances	108
8.46	arrAlphaID Struct Reference	108
8.46.1	Detailed Description	108
8.46.2	Field Documentation	108
8.46.2.1	allCallsAlphaIDInfoArr	108
8.46.2.2	numInstances	108
8.47	arrCalledPartyNum Struct Reference	108
8.47.1	Detailed Description	109
8.47.2	Field Documentation	109
8.47.2.1	CalledPartyNum	109
8.47.2.2	numInstances	109
8.48	arrCallEndReason Struct Reference	109
8.48.1	Detailed Description	109
8.48.2	Field Documentation	110
8.48.2.1	callEndReason	110
8.48.2.2	callID	110
8.48.2.3	numInstances	110
8.49	arrCallInfo Struct Reference	110
8.49.1	Detailed Description	110
8.49.2	Field Documentation	110
8.49.2.1	getAllCallInfo	110
8.49.2.2	numInstances	110
8.50	arrConnectPartyNum Struct Reference	110
8.50.1	Detailed Description	110
8.50.2	Field Documentation	111
8.50.2.1	ConnectedPartyNum	111
8.50.2.2	numInstances	111
8.51	arrDiagInfo Struct Reference	111
8.51.1	Detailed Description	111
8.51.2	Field Documentation	111
8.51.2.1	DiagInfo	111

8.51.2.2	numInstances	111
8.52	arrRedirPartyNum Struct Reference	111
8.52.1	Detailed Description	111
8.52.2	Field Documentation	112
8.52.2.1	numInstances	112
8.52.2.2	RedirPartyNum	112
8.53	arrRemotePartyName Struct Reference	112
8.53.1	Detailed Description	112
8.53.2	Field Documentation	112
8.53.2.1	GetAllCallRmtPtyName	112
8.53.2.2	numInstances	112
8.54	arrRemotePartyNum Struct Reference	112
8.54.1	Detailed Description	113
8.54.2	Field Documentation	113
8.54.2.1	numInstances	113
8.54.2.2	RmtPtyNum	113
8.55	arrSvcOption Struct Reference	113
8.55.1	Detailed Description	113
8.55.2	Field Documentation	113
8.55.2.1	callID	113
8.55.2.2	numInstances	113
8.55.2.3	srvOption	114
8.56	arrUUSInfo Struct Reference	114
8.56.1	Detailed Description	114
8.56.2	Field Documentation	114
8.56.2.1	AllCallsUUSInfo	114
8.56.2.2	numInstances	114
8.57	authenticateResult Struct Reference	114
8.57.1	Detailed Description	114
8.57.2	Field Documentation	115
8.57.2.1	content	115
8.57.2.2	contentLen	115
8.58	authenticationData Struct Reference	115
8.58.1	Detailed Description	115
8.58.2	Field Documentation	115
8.58.2.1	context	116
8.58.2.2	data	116
8.58.2.3	dataLen	116
8.59	BandCapabilityResp Struct Reference	116
8.59.1	Detailed Description	116

8.59.2	Field Documentation	119
8.59.2.1	bandCapability	119
8.59.2.2	pLteBandCapability	119
8.59.2.3	pTdsBandCapability	119
8.60	BdsSV Struct Reference	119
8.60.1	Detailed Description	119
8.60.2	Field Documentation	119
8.60.2.1	id	119
8.60.2.2	mask	119
8.61	BdsSVInfo Struct Reference	119
8.61.1	Detailed Description	120
8.61.2	Field Documentation	120
8.61.2.1	len	120
8.61.2.2	pSV	120
8.62	BroadcastConfig Struct Reference	120
8.62.1	Detailed Description	120
8.62.2	Field Documentation	121
8.62.2.1	fromServiceId	121
8.62.2.2	selected	121
8.62.2.3	toServiceId	121
8.63	burstDTMFInfo Struct Reference	121
8.63.1	Detailed Description	121
8.63.2	Field Documentation	121
8.63.2.1	digitCnt	121
8.63.2.2	pCallID	121
8.63.2.3	pDigitBuff	121
8.64	CallBarringSysInfo Struct Reference	121
8.64.1	Detailed Description	122
8.64.2	Field Documentation	122
8.64.2.1	csBarStatus	122
8.64.2.2	psBarStatus	122
8.65	callBarStatus Struct Reference	122
8.65.1	Detailed Description	122
8.65.2	Field Documentation	123
8.65.2.1	csBarStatus	123
8.65.2.2	psBarStatus	123
8.66	calledPartyInfo Struct Reference	123
8.66.1	Detailed Description	123
8.66.2	Field Documentation	124
8.66.2.1	number	124

8.66.2.2	numLen	124
8.66.2.3	numPlan	124
8.66.2.4	numType	125
8.66.2.5	PI	125
8.66.2.6	SI	125
8.67	calledPartySubAdd Struct Reference	125
8.67.1	Detailed Description	125
8.67.2	Field Documentation	125
8.67.2.1	extBit	125
8.67.2.2	oddEvenInd	125
8.67.2.3	subAddr	125
8.67.2.4	subAddrLen	125
8.67.2.5	subAddrType	126
8.68	callerIDInfo Struct Reference	126
8.68.1	Detailed Description	126
8.68.2	Field Documentation	126
8.68.2.1	callerID	126
8.68.2.2	callerIDLen	126
8.68.2.3	PI	126
8.69	callFwdTypeAndPlan Struct Reference	126
8.69.1	Detailed Description	126
8.69.2	Field Documentation	127
8.69.2.1	numberPlan	127
8.69.2.2	numberType	127
8.70	callFWExtInfo Struct Reference	127
8.70.1	Detailed Description	127
8.70.2	Field Documentation	129
8.70.2.1	noReplyTimer	129
8.70.2.2	number	129
8.70.2.3	numLen	129
8.70.2.4	numPlan	129
8.70.2.5	numType	129
8.70.2.6	PI	129
8.70.2.7	SI	129
8.70.2.8	SvcClass	129
8.70.2.9	SvcStatus	129
8.71	callFWInfo Struct Reference	129
8.71.1	Detailed Description	129
8.71.2	Field Documentation	130
8.71.2.1	noReplyTimer	130

8.71.2.2	number	130
8.71.2.3	numLen	130
8.71.2.4	SvcClass	130
8.71.2.5	SvcStatus	130
8.72	callInfo Struct Reference	130
8.72.1	Detailed Description	130
8.72.2	Field Documentation	131
8.72.2.1	callID	131
8.72.2.2	callState	132
8.72.2.3	callType	132
8.72.2.4	direction	132
8.72.2.5	mode	132
8.73	callingPartyInfo Struct Reference	132
8.73.1	Detailed Description	132
8.73.2	Field Documentation	133
8.73.2.1	number	133
8.73.2.2	numLen	133
8.73.2.3	numPlan	133
8.73.2.4	numType	133
8.73.2.5	PI	133
8.73.2.6	SI	133
8.74	cardResult Struct Reference	133
8.74.1	Detailed Description	134
8.74.2	Field Documentation	134
8.74.2.1	sw1	134
8.74.2.2	sw2	134
8.75	cardStatus Struct Reference	134
8.75.1	Detailed Description	134
8.75.2	Field Documentation	135
8.75.2.1	index1xPri	135
8.75.2.2	index1xSec	135
8.75.2.3	indexGwPri	135
8.75.2.4	indexGwSec	135
8.75.2.5	numSlot	135
8.75.2.6	SlotInfo	135
8.76	CarrierImage_t Struct Reference	135
8.76.1	Detailed Description	135
8.76.2	Field Documentation	136
8.76.2.1	m_FwBuildId	136
8.76.2.2	m_FwImageld	136

8.76.2.3	<a href="#">m_nCarrierId</a>	136
8.76.2.4	<a href="#">m_nFolderId</a>	136
8.76.2.5	<a href="#">m_nStorage</a>	136
8.76.2.6	<a href="#">m_PriBuildId</a>	136
8.76.2.7	<a href="#">m_PrImageld</a>	136
8.77	<a href="#">CatAlPhalIdentifierTlv Struct Reference</a>	136
8.77.1	<a href="#">Detailed Description</a>	137
8.77.2	<a href="#">Field Documentation</a>	137
8.77.2.1	<a href="#">AlphaID</a>	137
8.77.2.2	<a href="#">AlphaIDLength</a>	137
8.77.2.3	<a href="#">ReferenceID</a>	137
8.78	<a href="#">CatCommonEventTlv Struct Reference</a>	137
8.78.1	<a href="#">Field Documentation</a>	137
8.78.1.1	<a href="#">CatEvent</a>	137
8.78.1.2	<a href="#">EventID</a>	137
8.78.1.3	<a href="#">EventLength</a>	137
8.78.1.4	<a href="#">TlvPresent</a>	137
8.79	<a href="#">CatEndProactiveSessionTlv Struct Reference</a>	137
8.79.1	<a href="#">Detailed Description</a>	137
8.79.2	<a href="#">Field Documentation</a>	138
8.79.2.1	<a href="#">EndProactiveSession</a>	138
8.80	<a href="#">CATEventDataType Struct Reference</a>	138
8.80.1	<a href="#">Field Documentation</a>	138
8.80.1.1	<a href="#">eventMask</a>	138
8.80.1.2	<a href="#">pErrorMask</a>	138
8.81	<a href="#">CatEventIDDDataTlv Struct Reference</a>	138
8.81.1	<a href="#">Detailed Description</a>	138
8.81.2	<a href="#">Field Documentation</a>	138
8.81.2.1	<a href="#">Data</a>	138
8.81.2.2	<a href="#">DataLength</a>	138
8.81.2.3	<a href="#">ReferenceID</a>	138
8.82	<a href="#">CatEventListTlv Struct Reference</a>	138
8.82.1	<a href="#">Detailed Description</a>	139
8.82.2	<a href="#">Field Documentation</a>	139
8.82.2.1	<a href="#">SetupEventList</a>	139
8.83	<a href="#">CatRefreshTlv Struct Reference</a>	139
8.83.1	<a href="#">Detailed Description</a>	139
8.83.2	<a href="#">Field Documentation</a>	139
8.83.2.1	<a href="#">RefreshMode</a>	139
8.83.2.2	<a href="#">RefreshStage</a>	139

8.84	ccSUPSType Struct Reference	139
8.84.1	Detailed Description	140
8.84.2	Field Documentation	140
8.84.2.1	reason	140
8.84.2.2	svcType	140
8.85	CDMABroadcastConfig Struct Reference	140
8.85.1	Detailed Description	140
8.85.2	Field Documentation	141
8.85.2.1	language	141
8.85.2.2	selected	141
8.85.2.3	serviceCategory	141
8.86	CDMAChannel Struct Reference	141
8.86.1	Detailed Description	141
8.86.2	Field Documentation	141
8.86.2.1	priChA	141
8.86.2.2	priChB	141
8.86.2.3	secChA	142
8.86.2.4	secChB	142
8.87	CDMAECIOThresh Struct Reference	142
8.87.1	Detailed Description	142
8.87.2	Field Documentation	142
8.87.2.1	CDMAECIOThreshListLen	142
8.87.2.2	pCDMAECIOThreshList	142
8.88	CDMAInfo Struct Reference	142
8.88.1	Detailed Description	142
8.88.2	Field Documentation	143
8.88.2.1	baseId	143
8.88.2.2	baseLat	143
8.88.2.3	baseLong	143
8.88.2.4	nid	143
8.88.2.5	refpn	143
8.88.2.6	sid	143
8.89	cdmaMsgDecodingParams Struct Reference	143
8.89.1	Detailed Description	144
8.89.2	Field Documentation	145
8.89.2.1	absoluteValidity	145
8.89.2.2	mcTimeStamp	145
8.89.2.3	messageLength	145
8.89.2.4	pAlertPriority	145
8.89.2.5	pCallbkAddr	145

8.89.2.6	<a href="#">pCallbkAddrLength</a>	145
8.89.2.7	<a href="#">pDisplayMode</a>	145
8.89.2.8	<a href="#">pLanguage</a>	146
8.89.2.9	<a href="#">pMessage</a>	146
8.89.2.10	<a href="#">pMessageID</a>	146
8.89.2.11	<a href="#">pPriority</a>	146
8.89.2.12	<a href="#">pPrivacy</a>	146
8.89.2.13	<a href="#">pReadAcknowledgementReq</a>	146
8.89.2.14	<a href="#">pRelativeValidity</a>	146
8.89.2.15	<a href="#">pSenderAddr</a>	146
8.89.2.16	<a href="#">pSenderAddrLength</a>	146
8.89.2.17	<a href="#">pTextMsg</a>	146
8.89.2.18	<a href="#">pTextMsgLength</a>	146
8.89.2.19	<a href="#">pUserAcknowledgementReq</a>	146
8.90	<a href="#">cdmaMsgEncodingParams Struct Reference</a>	146
8.90.1	<a href="#">Detailed Description</a>	146
8.90.2	<a href="#">Field Documentation</a>	147
8.90.2.1	<a href="#">messageld</a>	147
8.90.2.2	<a href="#">pCallbackAddr</a>	147
8.90.2.3	<a href="#">pDestAddr</a>	147
8.90.2.4	<a href="#">pEncodingAlphabet</a>	147
8.90.2.5	<a href="#">pMessage</a>	147
8.90.2.6	<a href="#">pMessageSize</a>	147
8.90.2.7	<a href="#">pPriority</a>	147
8.90.2.8	<a href="#">pRelValidity</a>	148
8.90.2.9	<a href="#">pTextMsg</a>	148
8.90.2.10	<a href="#">textMsgLength</a>	148
8.91	<a href="#">CDMARSSIThresh Struct Reference</a>	148
8.91.1	<a href="#">Detailed Description</a>	148
8.91.2	<a href="#">Field Documentation</a>	148
8.91.2.1	<a href="#">CDMARSSIThreshListLen</a>	148
8.91.2.2	<a href="#">pCDMARSSIThreshList</a>	148
8.92	<a href="#">CDMASSInfo Struct Reference</a>	148
8.92.1	<a href="#">Detailed Description</a>	148
8.92.2	<a href="#">Field Documentation</a>	149
8.92.2.1	<a href="#">ecio</a>	149
8.92.2.2	<a href="#">rssi</a>	149
8.93	<a href="#">cdmaSSInfo Struct Reference</a>	149
8.93.1	<a href="#">Detailed Description</a>	149
8.93.2	<a href="#">Field Documentation</a>	149

8.93.2.1	ecio	149
8.93.2.2	rsi	149
8.94	CDMASysInfo Struct Reference	149
8.94.1	Detailed Description	150
8.94.2	Field Documentation	152
8.94.2.1	baseId	152
8.94.2.2	baseLat	152
8.94.2.3	baseLong	152
8.94.2.4	bsInfoValid	152
8.94.2.5	bsPRev	152
8.94.2.6	bsPRevValid	152
8.94.2.7	ccsSupported	152
8.94.2.8	ccsSupportedValid	152
8.94.2.9	cdmaSysIdValid	152
8.94.2.10	isSysPrIMatch	152
8.94.2.11	isSysPrIMatchValid	152
8.94.2.12	MCC	152
8.94.2.13	MNC	152
8.94.2.14	networkID	152
8.94.2.15	networkIdValid	153
8.94.2.16	packetZone	153
8.94.2.17	packetZoneValid	153
8.94.2.18	pRevInUse	153
8.94.2.19	pRevInUseValid	153
8.94.2.20	sysInfoCDMA	153
8.94.2.21	systemID	153
8.95	CDMASysInfoExt Struct Reference	153
8.95.1	Detailed Description	153
8.95.2	Field Documentation	153
8.95.2.1	imsi_11_12	153
8.95.2.2	MCC	153
8.96	CellIDb Struct Reference	153
8.96.1	Detailed Description	153
8.96.2	Field Documentation	154
8.96.2.1	mask	154
8.97	cellParams Struct Reference	154
8.97.1	Detailed Description	154
8.97.2	Field Documentation	155
8.97.2.1	pci	155
8.97.2.2	rsrp	155

8.97.2.3	rsrq	155
8.97.2.4	rssi	155
8.97.2.5	srxlev	155
8.98	changeUIMPIN Struct Reference	155
8.98.1	Detailed Description	155
8.98.2	Field Documentation	156
8.98.2.1	oldPINLen	156
8.98.2.2	oldPINVal	156
8.98.2.3	pinID	156
8.98.2.4	pinLen	156
8.98.2.5	pinVal	156
8.99	ChannelRate Struct Reference	156
8.99.1	Detailed Description	156
8.99.2	Field Documentation	156
8.99.2.1	CurrChanRxRate	156
8.99.2.2	CurrChanTxRate	156
8.99.2.3	MaxChanRxRate	156
8.99.2.4	MaxChanTxRate	157
8.100	channelRate Struct Reference	157
8.100.1	Detailed Description	157
8.100.2	Field Documentation	157
8.100.2.1	CurrChanRxRate	157
8.100.2.2	CurrChanTxRate	157
8.101	CLIPResp Struct Reference	157
8.101.1	Detailed Description	157
8.101.2	Field Documentation	158
8.101.2.1	ActiveStatus	158
8.101.2.2	ProvisionStatus	158
8.102	CLIRResp Struct Reference	158
8.102.1	Detailed Description	158
8.102.2	Field Documentation	158
8.102.2.1	ActiveStatus	158
8.102.2.2	ProvisionStatus	158
8.103	CIkInfo Struct Reference	159
8.103.1	Detailed Description	159
8.103.2	Field Documentation	160
8.103.2.1	mask	160
8.104	CNAPResp Struct Reference	160
8.104.1	Detailed Description	160
8.104.2	Field Documentation	160

8.104.2.1	ActiveStatus	160
8.104.2.2	ProvisionStatus	160
8.105	COLPResp Struct Reference	160
8.105.1	Detailed Description	160
8.105.2	Field Documentation	161
8.105.2.1	ActiveStatus	161
8.105.2.2	ProvisionStatus	161
8.106	COLRResp Struct Reference	161
8.106.1	Detailed Description	161
8.106.2	Field Documentation	162
8.106.2.1	ActiveStatus	162
8.106.2.2	ProvisionStatus	162
8.107	CommInfo Struct Reference	162
8.107.1	Detailed Description	162
8.107.2	Field Documentation	163
8.107.2.1	imsRegState	163
8.107.2.2	modemMode	163
8.107.2.3	psState	163
8.107.2.4	systemMode	163
8.107.2.5	temperature	163
8.108	ConnectionStatus Struct Reference	163
8.108.1	Detailed Description	164
8.108.2	Field Documentation	164
8.108.2.1	MDMCallDuration	164
8.108.2.2	MDMConnStatus	164
8.109	connectionStatus Struct Reference	164
8.109.1	Detailed Description	164
8.109.2	Field Documentation	164
8.109.2.1	MDMCallDuration	164
8.109.2.2	MDMConnStatus	164
8.110	connectNumInfo Struct Reference	164
8.110.1	Detailed Description	165
8.110.2	Field Documentation	166
8.110.2.1	callerID	166
8.110.2.2	callerIDLen	166
8.110.2.3	numPlan	166
8.110.2.4	numPresInd	166
8.110.2.5	numType	166
8.110.2.6	screeningInd	166
8.111	CrashInfo Struct Reference	166

8.111.1 Detailed Description . . . . .	166
8.111.2 Field Documentation . . . . .	167
8.111.2.1 crashData . . . . .	167
8.111.2.2 crashId . . . . .	167
8.111.2.3 crashStrLen . . . . .	167
8.111.2.4 gcDumpStrLen . . . . .	167
8.111.2.5 numCrashes . . . . .	167
8.111.2.6 pCrashString . . . . .	167
8.111.2.7 pGCDumpString . . . . .	167
8.112 CrashInfoParams Struct Reference . . . . .	167
8.112.1 Detailed Description . . . . .	167
8.112.2 Field Documentation . . . . .	168
8.112.2.1 pCrashInfo . . . . .	168
8.112.2.2 pDevCrashStatus . . . . .	168
8.113 CreateProfileIn Struct Reference . . . . .	168
8.113.1 Detailed Description . . . . .	168
8.113.2 Field Documentation . . . . .	168
8.113.2.1 curProfile . . . . .	169
8.113.2.2 pProfileID . . . . .	169
8.113.2.3 pProfileType . . . . .	169
8.114 CreateProfileOut Struct Reference . . . . .	169
8.114.1 Detailed Description . . . . .	169
8.114.2 Field Documentation . . . . .	169
8.114.2.1 pExtErrorCode . . . . .	169
8.114.2.2 pProfileIndex . . . . .	169
8.114.2.3 pProfileType . . . . .	169
8.115 CSGID Struct Reference . . . . .	169
8.115.1 Detailed Description . . . . .	169
8.115.2 Field Documentation . . . . .	170
8.115.2.1 id . . . . .	170
8.115.2.2 mcc . . . . .	170
8.115.2.3 mnc . . . . .	170
8.115.2.4 mncPcsDigits . . . . .	170
8.115.2.5 rat . . . . .	170
8.116 CUGInfo Struct Reference . . . . .	170
8.116.1 Detailed Description . . . . .	170
8.116.2 Field Documentation . . . . .	171
8.116.2.1 CUGIndex . . . . .	171
8.116.2.2 SuppOA . . . . .	171
8.116.2.3 SuppPrefCUG . . . . .	171

8.117curAMRConfig Struct Reference . . . . .	171
8.117.1 Detailed Description . . . . .	171
8.117.2 Field Documentation . . . . .	171
8.117.2.1 gsmAmrStat . . . . .	172
8.117.2.2 wcdmaAmrStat . . . . .	172
8.118CurrDataSysStat Struct Reference . . . . .	172
8.118.1 Detailed Description . . . . .	172
8.118.2 Field Documentation . . . . .	172
8.118.2.1 pCurrNetworkInfo . . . . .	172
8.118.2.2 pNetworkInfoLen . . . . .	172
8.118.2.3 pPrefNetwork . . . . .	172
8.119currentCatEvent Union Reference . . . . .	172
8.119.1 Detailed Description . . . . .	173
8.119.2 Field Documentation . . . . .	173
8.119.2.1 CatAlphaldtfr . . . . .	173
8.119.2.2 CatEndPS . . . . .	173
8.119.2.3 CatEventLst . . . . .	173
8.119.2.4 CatEvIDData . . . . .	173
8.119.2.5 CatRefresh . . . . .	173
8.120CurrentImglList Struct Reference . . . . .	173
8.120.1 Detailed Description . . . . .	173
8.120.2 Field Documentation . . . . .	174
8.120.2.1 carrier . . . . .	174
8.120.2.2 fwvers . . . . .	174
8.120.2.3 numEntries . . . . .	174
8.120.2.4 pCurrImglInfo . . . . .	174
8.120.2.5 pkgver . . . . .	174
8.120.2.6 priver . . . . .	174
8.121currentPLMN Struct Reference . . . . .	174
8.121.1 Detailed Description . . . . .	174
8.121.2 Field Documentation . . . . .	175
8.121.2.1 MCC . . . . .	175
8.121.2.2 MNC . . . . .	175
8.121.2.3 netDescr . . . . .	175
8.121.2.4 netDescrLength . . . . .	175
8.122CurrImageInfo Struct Reference . . . . .	175
8.122.1 Detailed Description . . . . .	175
8.122.2 Field Documentation . . . . .	176
8.122.2.1 buildID . . . . .	176
8.122.2.2 buildIDLen . . . . .	176

8.122.2.3 imageType . . . . .	176
8.122.2.4 uniqueID . . . . .	176
8.123CurrNetworkInfo Struct Reference . . . . .	176
8.123.1 Detailed Description . . . . .	176
8.123.2 Field Documentation . . . . .	178
8.123.2.1 NetworkType . . . . .	178
8.123.2.2 RATMask . . . . .	178
8.123.2.3 SOMask . . . . .	178
8.124currNetworkInfo Struct Reference . . . . .	178
8.124.1 Detailed Description . . . . .	178
8.124.2 Field Documentation . . . . .	178
8.124.2.1 NetworkType . . . . .	179
8.124.2.2 RATMask . . . . .	179
8.124.2.3 SOMask . . . . .	179
8.125custFeaturesInfo Struct Reference . . . . .	179
8.125.1 Detailed Description . . . . .	179
8.125.2 Field Documentation . . . . .	181
8.125.2.1 GpsEnable . . . . .	181
8.125.2.2 pDHCPRelayEnabled . . . . .	181
8.125.2.3 pDisableIMSI . . . . .	181
8.125.2.4 pGPSLPM . . . . .	181
8.125.2.5 pGPSSel . . . . .	181
8.125.2.6 pIPFamSupport . . . . .	181
8.125.2.7 plsVoiceEnabled . . . . .	181
8.125.2.8 pRMAutoConnect . . . . .	181
8.125.2.9 pSMSSupport . . . . .	181
8.126custFeaturesSetting Struct Reference . . . . .	181
8.126.1 Detailed Description . . . . .	181
8.126.2 Field Documentation . . . . .	183
8.126.2.1 pDHCPRelayEnabled . . . . .	183
8.126.2.2 pGPSEnable . . . . .	183
8.126.2.3 pGPSLPM . . . . .	183
8.126.2.4 pGPSSel . . . . .	183
8.126.2.5 plsVoiceEnabled . . . . .	183
8.127custSettingInfo Struct Reference . . . . .	183
8.127.1 Detailed Description . . . . .	183
8.127.2 Field Documentation . . . . .	183
8.127.2.1 cust_attr . . . . .	183
8.127.2.2 cust_id . . . . .	183
8.127.2.3 cust_value . . . . .	184

8.127.2.4 id_length . . . . .	184
8.127.2.5 value_length . . . . .	184
8.128custSettingList Struct Reference . . . . .	184
8.128.1 Detailed Description . . . . .	184
8.128.2 Field Documentation . . . . .	184
8.128.2.1 custSetting . . . . .	184
8.128.2.2 list_type . . . . .	184
8.128.2.3 num_instances . . . . .	184
8.129dataBearers Struct Reference . . . . .	184
8.129.1 Detailed Description . . . . .	184
8.129.2 Field Documentation . . . . .	185
8.129.2.1 dataBearerMask . . . . .	185
8.129.2.2 pCurDataBearerTechnology . . . . .	185
8.129.2.3 pLastCallDataBearerTechnology . . . . .	185
8.130DataBearerTech Struct Reference . . . . .	185
8.130.1 Detailed Description . . . . .	185
8.130.2 Field Documentation . . . . .	187
8.130.2.1 ratValue . . . . .	187
8.130.2.2 soMask . . . . .	187
8.130.2.3 techType . . . . .	187
8.131DataBearerTechExt Struct Reference . . . . .	187
8.131.1 Detailed Description . . . . .	188
8.131.2 Field Documentation . . . . .	188
8.131.2.1 pBearerTech . . . . .	188
8.131.2.2 pLastBearerTech . . . . .	188
8.132dataBearerTechnology Struct Reference . . . . .	188
8.132.1 Detailed Description . . . . .	188
8.132.2 Field Documentation . . . . .	189
8.132.2.1 currentNetwork . . . . .	189
8.132.2.2 ratMask . . . . .	189
8.132.2.3 soMask . . . . .	189
8.133dataRate Struct Reference . . . . .	189
8.133.1 Detailed Description . . . . .	189
8.133.2 Field Documentation . . . . .	190
8.133.2.1 dataRateMax . . . . .	190
8.133.2.2 guaranteedRate . . . . .	190
8.134dataSrvCapabilities Struct Reference . . . . .	190
8.134.1 Detailed Description . . . . .	190
8.134.2 Field Documentation . . . . .	190
8.134.2.1 dataCapabilities . . . . .	190

8.134.2.2 dataCapabilitiesLen . . . . .	191
8.135DataStatusDetail Struct Reference . . . . .	191
8.135.1 Detailed Description . . . . .	191
8.135.2 Field Documentation . . . . .	192
8.135.2.1 IPAddress . . . . .	192
8.135.2.2 LastErrCode . . . . .	192
8.136DataULongLongTlv Struct Reference . . . . .	193
8.136.1 Field Documentation . . . . .	193
8.136.1.1 TlvPresent . . . . .	193
8.136.1.2 ullData . . . . .	193
8.137DataULongTlv Struct Reference . . . . .	193
8.137.1 Field Documentation . . . . .	193
8.137.1.1 TlvPresent . . . . .	193
8.137.1.2 ulData . . . . .	193
8.138DcsUsbPortNames Struct Reference . . . . .	193
8.138.1 Field Documentation . . . . .	193
8.138.1.1 AtCmdPort . . . . .	193
8.138.1.2 DmPort . . . . .	193
8.138.1.3 NmeaPort . . . . .	193
8.139delAssistDataStatus Struct Reference . . . . .	193
8.139.1 Detailed Description . . . . .	194
8.139.2 Field Documentation . . . . .	194
8.139.2.1 status . . . . .	194
8.140depersonalizationInformation Struct Reference . . . . .	194
8.140.1 Detailed Description . . . . .	194
8.140.2 Field Documentation . . . . .	195
8.140.2.1 ckLen . . . . .	195
8.140.2.2 ckVal . . . . .	195
8.140.2.3 feature . . . . .	195
8.140.2.4 operation . . . . .	195
8.141detailSvcInfo Struct Reference . . . . .	195
8.141.1 Detailed Description . . . . .	195
8.141.2 Field Documentation . . . . .	196
8.141.2.1 hdrHybrid . . . . .	196
8.141.2.2 hdrSrvStatus . . . . .	196
8.141.2.3 isSysForbidden . . . . .	196
8.141.2.4 srvCapability . . . . .	197
8.141.2.5 srvStatus . . . . .	197
8.142DeviceConfigDetail Struct Reference . . . . .	197
8.142.1 Detailed Description . . . . .	197

8.142.2 Field Documentation . . . . .	197
8.142.2.1 Chipset . . . . .	197
8.142.2.2 HWVersion . . . . .	198
8.142.2.3 QLIC . . . . .	198
8.142.2.4 Technology . . . . .	198
8.143DHCPOption Struct Reference . . . . .	198
8.143.1 Detailed Description . . . . .	198
8.143.2 Field Documentation . . . . .	198
8.143.2.1 optCode . . . . .	198
8.143.2.2 optValLen . . . . .	198
8.143.2.3 pOptVal . . . . .	198
8.144DHCPOptionList Struct Reference . . . . .	198
8.144.1 Detailed Description . . . . .	198
8.144.2 Field Documentation . . . . .	199
8.144.2.1 numOpt . . . . .	199
8.144.2.2 pOptions . . . . .	199
8.145diagInfo Struct Reference . . . . .	199
8.145.1 Detailed Description . . . . .	199
8.145.2 Field Documentation . . . . .	199
8.145.2.1 diagInfoLen . . . . .	199
8.145.2.2 diagnosticInfo . . . . .	199
8.146dirNum Struct Reference . . . . .	199
8.146.1 Detailed Description . . . . .	200
8.146.2 Field Documentation . . . . .	200
8.146.2.1 dirNum . . . . .	200
8.146.2.2 dirNumLen . . . . .	200
8.147dms_ActivationStatusTlv Struct Reference . . . . .	200
8.147.1 Detailed Description . . . . .	200
8.147.2 Field Documentation . . . . .	200
8.147.2.1 activationStatus . . . . .	200
8.147.2.2 TlvPresent . . . . .	201
8.148dms_OperatingModeTlv Struct Reference . . . . .	201
8.148.1 Detailed Description . . . . .	201
8.148.2 Field Documentation . . . . .	201
8.148.2.1 operatingMode . . . . .	201
8.148.2.2 TlvPresent . . . . .	201
8.149dmsCurrentPRLInfo Struct Reference . . . . .	201
8.149.1 Detailed Description . . . . .	201
8.149.2 Field Documentation . . . . .	202
8.149.2.1 pPRLPreference . . . . .	202

8.149.2.2 pPRLVersion . . . . .	202
8.150DMScustSettingInfo Struct Reference . . . . .	202
8.150.1 Detailed Description . . . . .	202
8.150.2 Field Documentation . . . . .	202
8.150.2.1 cust_attr . . . . .	202
8.150.2.2 cust_id . . . . .	202
8.150.2.3 cust_value . . . . .	202
8.150.2.4 id_length . . . . .	202
8.150.2.5 value_length . . . . .	202
8.151DMScustSettingList Struct Reference . . . . .	203
8.151.1 Detailed Description . . . . .	203
8.151.2 Field Documentation . . . . .	203
8.151.2.1 custSetting . . . . .	203
8.151.2.2 list_type . . . . .	203
8.151.2.3 num_instances . . . . .	203
8.152DMSgetCustomFeatureV2 Struct Reference . . . . .	203
8.152.1 Detailed Description . . . . .	203
8.152.2 Field Documentation . . . . .	204
8.152.2.1 pCustSettingInfo . . . . .	204
8.152.2.2 pCustSettingList . . . . .	204
8.152.2.3 pGetCustomInput . . . . .	204
8.153DMSgetCustomInput Struct Reference . . . . .	204
8.153.1 Detailed Description . . . . .	204
8.153.2 Field Documentation . . . . .	204
8.153.2.1 cust_id . . . . .	204
8.153.2.2 list_type . . . . .	204
8.154dmsIndicationRegisterReq Struct Reference . . . . .	204
8.154.1 Detailed Description . . . . .	204
8.154.2 Field Documentation . . . . .	205
8.154.2.1 pSwiGetResetInd . . . . .	205
8.155dmsSwiGetResetInfo Struct Reference . . . . .	205
8.155.1 Detailed Description . . . . .	205
8.155.2 Field Documentation . . . . .	205
8.155.2.1 source . . . . .	206
8.155.2.2 type . . . . .	206
8.156Domain Struct Reference . . . . .	206
8.156.1 Detailed Description . . . . .	206
8.156.2 Field Documentation . . . . .	206
8.156.2.1 domainLen . . . . .	206
8.156.2.2 domainName . . . . .	206

8.157DomainNameList Struct Reference . . . . .	206
8.157.1 Detailed Description . . . . .	206
8.157.2 Field Documentation . . . . .	206
8.157.2.1 domain . . . . .	207
8.157.2.2 numInstances . . . . .	207
8.158DRCPParams Struct Reference . . . . .	207
8.158.1 Detailed Description . . . . .	207
8.158.2 Field Documentation . . . . .	207
8.158.2.1 DRCCover . . . . .	207
8.158.2.2 DRCValue . . . . .	207
8.159DTMFInfo Struct Reference . . . . .	207
8.159.1 Detailed Description . . . . .	207
8.159.2 Field Documentation . . . . .	208
8.159.2.1 callID . . . . .	208
8.159.2.2 digitBuff . . . . .	208
8.159.2.3 digitCnt . . . . .	208
8.159.2.4 DTMFEvent . . . . .	208
8.160DTMFLengths Struct Reference . . . . .	208
8.160.1 Detailed Description . . . . .	208
8.160.2 Field Documentation . . . . .	209
8.160.2.1 DTMFInterdigitInterval . . . . .	209
8.160.2.2 DTMFPulseWidth . . . . .	209
8.161DUNCallInfoInd Struct Reference . . . . .	209
8.161.1 Field Documentation . . . . .	209
8.161.1.1 CallEndReason . . . . .	209
8.161.1.2 ChannelRate . . . . .	209
8.161.1.3 DataBearerTech . . . . .	209
8.161.1.4 DormancyStatus . . . . .	209
8.161.1.5 MdmConnStatus . . . . .	209
8.161.1.6 RXOKBytesCount . . . . .	209
8.161.1.7 TXOKBytesCount . . . . .	209
8.162dunchannelRate Struct Reference . . . . .	210
8.162.1 Detailed Description . . . . .	210
8.162.2 Field Documentation . . . . .	210
8.162.2.1 CurrChanRxRate . . . . .	210
8.162.2.2 CurrChanTxRate . . . . .	210
8.162.2.3 MaxChanRxRate . . . . .	210
8.162.2.4 MaxChanTxRate . . . . .	210
8.163ecioListElement Struct Reference . . . . .	210
8.163.1 Detailed Description . . . . .	210

8.163.2 Field Documentation . . . . .	211
8.163.2.1 ecio . . . . .	211
8.163.2.2 radiolf . . . . .	211
8.164ECIOTresh Struct Reference . . . . .	211
8.164.1 Detailed Description . . . . .	211
8.164.2 Field Documentation . . . . .	211
8.164.2.1 ECIOThresListLen . . . . .	211
8.164.2.2 pECIOThresList . . . . .	212
8.165ECTNum Struct Reference . . . . .	212
8.165.1 Detailed Description . . . . .	212
8.165.2 Field Documentation . . . . .	212
8.165.2.1 ECTCallState . . . . .	212
8.165.2.2 number . . . . .	212
8.165.2.3 presentationInd . . . . .	212
8.166encryptedPIN1 Struct Reference . . . . .	212
8.166.1 Detailed Description . . . . .	212
8.166.2 Field Documentation . . . . .	213
8.166.2.1 pin1Len . . . . .	213
8.166.2.2 pin1Val . . . . .	213
8.167ERFileparams Struct Reference . . . . .	213
8.167.1 Detailed Description . . . . .	213
8.167.2 Field Documentation . . . . .	213
8.167.2.1 pFile . . . . .	213
8.167.2.2 pFileSize . . . . .	213
8.168errorRateListElement Struct Reference . . . . .	213
8.168.1 Detailed Description . . . . .	214
8.168.2 Field Documentation . . . . .	214
8.168.2.1 errorRate . . . . .	214
8.168.2.2 radiolf . . . . .	214
8.169eTWSPLMNInfoTlv Struct Reference . . . . .	214
8.169.1 Detailed Description . . . . .	214
8.169.2 Field Documentation . . . . .	215
8.169.2.1 ETWSPLMNInfo . . . . .	215
8.169.2.2 TlvPresent . . . . .	215
8.170extDispRecInfo Struct Reference . . . . .	215
8.170.1 Detailed Description . . . . .	215
8.170.2 Field Documentation . . . . .	215
8.170.2.1 dispType . . . . .	215
8.170.2.2 extDispInfo . . . . .	215
8.170.2.3 extDispInfoLen . . . . .	215

8.171 FactorySequenceNumber Struct Reference . . . . .	215
8.171.1 Detailed Description . . . . .	216
8.171.2 Field Documentation . . . . .	216
8.171.2.1 FSNumber . . . . .	216
8.172 fileAttributes Struct Reference . . . . .	216
8.172.1 Detailed Description . . . . .	216
8.172.2 Field Documentation . . . . .	219
8.172.2.1 fileID . . . . .	219
8.172.2.2 fileSize . . . . .	219
8.172.2.3 fileType . . . . .	219
8.172.2.4 rawLen . . . . .	219
8.172.2.5 rawValue . . . . .	219
8.172.2.6 recordCount . . . . .	219
8.172.2.7 recordSize . . . . .	219
8.172.2.8 secActivate . . . . .	219
8.172.2.9 secActivateMask . . . . .	219
8.172.2.10 secDeactivate . . . . .	219
8.172.2.11 secDeactivateMask . . . . .	219
8.172.2.12 secIncrease . . . . .	219
8.172.2.13 secIncreaseMask . . . . .	219
8.172.2.14 secRead . . . . .	219
8.172.2.15 secReadMask . . . . .	219
8.172.2.16 secWrite . . . . .	219
8.172.2.17 secWriteMask . . . . .	219
8.173 fileInfo Struct Reference . . . . .	219
8.173.1 Detailed Description . . . . .	219
8.173.2 Field Documentation . . . . .	220
8.173.2.1 fileID . . . . .	220
8.173.2.2 path . . . . .	220
8.173.2.3 pathLen . . . . .	220
8.174 FirmwareUpdatStat Struct Reference . . . . .	220
8.174.1 Detailed Description . . . . .	220
8.174.2 Field Documentation . . . . .	221
8.174.2.1 plmgType . . . . .	221
8.174.2.2 pLogString . . . . .	222
8.174.2.3 pLogStringLen . . . . .	222
8.174.2.4 pRefData . . . . .	222
8.174.2.5 pRefString . . . . .	222
8.174.2.6 pRefStringLen . . . . .	222
8.174.2.7 ResCode . . . . .	222

8.175FMSImageElement Struct Reference . . . . .	222
8.175.1 Detailed Description . . . . .	222
8.175.2 Field Documentation . . . . .	222
8.175.2.1 buildId . . . . .	222
8.175.2.2 buildIdLength . . . . .	222
8.175.2.3 imageId . . . . .	222
8.175.2.4 imageType . . . . .	222
8.176FMSImageIdElement Struct Reference . . . . .	222
8.176.1 Detailed Description . . . . .	223
8.176.2 Field Documentation . . . . .	223
8.176.2.1 buildID . . . . .	223
8.176.2.2 buildIDLength . . . . .	223
8.176.2.3 failureCount . . . . .	223
8.176.2.4 imageID . . . . .	223
8.176.2.5 storageIndex . . . . .	223
8.177FMSImageIDEntries Struct Reference . . . . .	223
8.177.1 Detailed Description . . . . .	224
8.177.2 Field Documentation . . . . .	224
8.177.2.1 executingImage . . . . .	224
8.177.2.2 imageIDElement . . . . .	224
8.177.2.3 imageIDSize . . . . .	224
8.177.2.4 imageType . . . . .	224
8.177.2.5 maxImages . . . . .	224
8.178FMSImageList Struct Reference . . . . .	224
8.178.1 Detailed Description . . . . .	224
8.178.2 Field Documentation . . . . .	225
8.178.2.1 imageIDEntries . . . . .	225
8.178.2.2 listSize . . . . .	225
8.179FMSPrefImageList Struct Reference . . . . .	225
8.179.1 Detailed Description . . . . .	225
8.179.2 Field Documentation . . . . .	225
8.179.2.1 listEntries . . . . .	225
8.179.2.2 listSize . . . . .	225
8.180fwinfo_s Struct Reference . . . . .	225
8.180.1 Detailed Description . . . . .	225
8.180.2 Field Documentation . . . . .	226
8.180.2.1 Carrier . . . . .	226
8.180.2.2 FirmwareID . . . . .	226
8.180.2.3 GPSCapability . . . . .	226
8.180.2.4 Region . . . . .	226

8.180.2.5 Technology	226
8.181 GERANInfo Struct Reference	226
8.181.1 Detailed Description	226
8.181.2 Field Documentation	227
8.181.2.1 arfcn	227
8.181.2.2 bsic	227
8.181.2.3 cellID	227
8.181.2.4 insNmrCellInfo	228
8.181.2.5 lac	228
8.181.2.6 nmrlnst	228
8.181.2.7 plmn	228
8.181.2.8 rxLev	228
8.181.2.9 timingAdvance	228
8.182 geranInstInfo Struct Reference	228
8.182.1 Detailed Description	228
8.182.2 Field Documentation	228
8.182.2.1 geranArfcn	228
8.182.2.2 geranBsicBcc	228
8.182.2.3 geranBsicNcc	228
8.182.2.4 geranRssi	228
8.183 getAllCallInformation Struct Reference	229
8.183.1 Detailed Description	229
8.183.2 Field Documentation	229
8.183.2.1 ALS	229
8.183.2.2 Callinfo	229
8.183.2.3 isEmpty	229
8.184 getAllCallRmtPtyName Struct Reference	229
8.184.1 Detailed Description	229
8.184.2 Field Documentation	230
8.184.2.1 callID	230
8.184.2.2 RemotePartyName	230
8.185 getAllCallRmtPtyNum Struct Reference	230
8.185.1 Detailed Description	230
8.185.2 Field Documentation	230
8.185.2.1 callID	230
8.185.2.2 RemotePartyNum	230
8.186 GetAudioPathConfigReq Struct Reference	230
8.186.1 Detailed Description	230
8.186.2 Field Documentation	231
8.186.2.1 Item	231

8.186.2.2 Profile . . . . .	231
8.187GetAudioPathConfigResp Struct Reference . . . . .	231
8.187.1 Detailed Description . . . . .	231
8.187.2 Field Documentation . . . . .	232
8.187.2.1 pCodecSTGain . . . . .	233
8.187.2.2 pDTMFTXGain . . . . .	233
8.187.2.3 pECMode . . . . .	233
8.187.2.4 pMICGainSelect . . . . .	233
8.187.2.5 pNSEnable . . . . .	233
8.187.2.6 pRXAGCList . . . . .	233
8.187.2.7 pRXAVCAGCSwitch . . . . .	233
8.187.2.8 pRXAVCList . . . . .	233
8.187.2.9 pRXPCMIIRFitr . . . . .	233
8.187.2.10pTXAGCList . . . . .	233
8.187.2.11pTXAVCSwitch . . . . .	233
8.187.2.12pTXGain . . . . .	233
8.187.2.13pTXPCMIIRFitr . . . . .	233
8.188GetAudioProfileReq Struct Reference . . . . .	233
8.188.1 Detailed Description . . . . .	233
8.188.2 Field Documentation . . . . .	233
8.188.2.1 Generator . . . . .	233
8.189GetAudioProfileResp Struct Reference . . . . .	233
8.189.1 Detailed Description . . . . .	234
8.189.2 Field Documentation . . . . .	234
8.189.2.1 EarMute . . . . .	234
8.189.2.2 MicMute . . . . .	234
8.189.2.3 Profile . . . . .	234
8.189.2.4 Volume . . . . .	234
8.190GetAudioVoTLBConfigReq Struct Reference . . . . .	235
8.190.1 Detailed Description . . . . .	235
8.190.2 Field Documentation . . . . .	235
8.190.2.1 Generator . . . . .	235
8.190.2.2 Item . . . . .	235
8.190.2.3 Profile . . . . .	235
8.190.2.4 Volume . . . . .	235
8.191GetAudioVoTLBConfigResp Struct Reference . . . . .	235
8.191.1 Detailed Description . . . . .	236
8.191.2 Field Documentation . . . . .	236
8.191.2.1 ResCode . . . . .	236
8.192getCallFWExtInfo Struct Reference . . . . .	236

8.192.1 Detailed Description . . . . .	236
8.192.2 Field Documentation . . . . .	236
8.192.2.1 CallFWExtInfo . . . . .	236
8.192.2.2 numInstances . . . . .	236
8.193getCallFWInfo Struct Reference . . . . .	236
8.193.1 Detailed Description . . . . .	236
8.193.2 Field Documentation . . . . .	237
8.193.2.1 CallFWInfo . . . . .	237
8.193.2.2 numInstances . . . . .	237
8.194getCustomFeatureV2 Struct Reference . . . . .	237
8.194.1 Detailed Description . . . . .	237
8.194.2 Field Documentation . . . . .	237
8.194.2.1 pCustSettingInfo . . . . .	237
8.194.2.2 pCustSettingList . . . . .	237
8.194.2.3 pGetCustomInput . . . . .	237
8.195getCustomInput Struct Reference . . . . .	238
8.195.1 Detailed Description . . . . .	238
8.195.2 Field Documentation . . . . .	238
8.195.2.1 cust_id . . . . .	238
8.195.2.2 list_type . . . . .	238
8.196getDUNCallInfoReq Struct Reference . . . . .	238
8.196.1 Detailed Description . . . . .	238
8.196.2 Field Documentation . . . . .	239
8.196.2.1 Mask . . . . .	239
8.196.2.2 pReportChannelRate . . . . .	239
8.196.2.3 pReportConnStatus . . . . .	239
8.196.2.4 pReportDataBearerTech . . . . .	239
8.196.2.5 pReportDormStatus . . . . .	239
8.196.2.6 pTransferStatInd . . . . .	239
8.197getDUNCallInfoResp Struct Reference . . . . .	240
8.197.1 Detailed Description . . . . .	240
8.197.2 Field Documentation . . . . .	242
8.197.2.1 pCallEndReason . . . . .	242
8.197.2.2 pChannelRate . . . . .	242
8.197.2.3 pConnectionStatus . . . . .	242
8.197.2.4 pDataBearerTech . . . . .	242
8.197.2.5 pDormancyStatus . . . . .	242
8.197.2.6 pLastCallDataBearerTech . . . . .	242
8.197.2.7 pLastCallRXOKBytesCnt . . . . .	242
8.197.2.8 pLastCallTXOKBytesCnt . . . . .	242

8.197.2.9 pMdmCallDurationActive . . . . .	242
8.197.2.10pRXOKBytesCount . . . . .	242
8.197.2.11pTXOKBytesCount . . . . .	242
8.198getDyingGaspCfg Struct Reference . . . . .	242
8.198.1 Detailed Description . . . . .	243
8.198.2 Field Documentation . . . . .	243
8.198.2.1 pDestSMSContent . . . . .	243
8.198.2.2 pDestSMSNum . . . . .	243
8.199getDyingGaspStatistics Struct Reference . . . . .	243
8.199.1 Detailed Description . . . . .	243
8.199.2 Field Documentation . . . . .	243
8.199.2.1 pSMSAttemptedFlag . . . . .	243
8.199.2.2 pTimeStamp . . . . .	244
8.200GetErrRateResp Struct Reference . . . . .	244
8.200.1 Detailed Description . . . . .	244
8.200.2 Field Documentation . . . . .	244
8.200.2.1 pCDMAFrameErrRate . . . . .	244
8.200.2.2 pGSMBER . . . . .	244
8.200.2.3 pHDRPackErrRate . . . . .	244
8.200.2.4 pWCDMABER . . . . .	244
8.201GetHRPDStatsResp Struct Reference . . . . .	244
8.201.1 Detailed Description . . . . .	245
8.201.2 Field Documentation . . . . .	245
8.201.2.1 pDRCPParams . . . . .	245
8.201.2.2 pPilotSetData . . . . .	245
8.201.2.3 pUATI . . . . .	245
8.202GetIMSSMSConfigParams Struct Reference . . . . .	245
8.202.1 Detailed Description . . . . .	245
8.202.2 Field Documentation . . . . .	246
8.202.2.1 pPhoneCtxtURI . . . . .	246
8.202.2.2 pPhoneCtxtURILen . . . . .	246
8.202.2.3 pSettingResp . . . . .	246
8.202.2.4 pSMSFormat . . . . .	246
8.202.2.5 pSMSOverIPNwInd . . . . .	246
8.203GetIMSUserConfigParams Struct Reference . . . . .	246
8.203.1 Detailed Description . . . . .	246
8.203.2 Field Documentation . . . . .	247
8.203.2.1 pIMSDomain . . . . .	247
8.203.2.2 pIMSDomainLen . . . . .	247
8.203.2.3 pSettingResp . . . . .	247

8.204GetIMSVoIPConfigResp Struct Reference	247
8.204.1 Detailed Description	247
8.204.2 Field Documentation	249
8.204.2.1 pAmrMode	249
8.204.2.2 pAmrOctetAligned	249
8.204.2.3 pAmrWbEnable	249
8.204.2.4 pAmrWBMode	249
8.204.2.5 pAmrWBOctetAligned	249
8.204.2.6 pMinSessionExpiryTimer	249
8.204.2.7 pRingBackTimer	249
8.204.2.8 pRingingTimer	249
8.204.2.9 pRTPRTCPInactTimer	249
8.204.2.10pScrAmrEnable	249
8.204.2.11pScrAmrWbEnable	249
8.204.2.12pSessionExpiryTimer	249
8.204.2.13pSettingResp	249
8.205GetInstIDResp Struct Reference	249
8.205.1 Field Documentation	249
8.205.1.1 pInstanceID	249
8.205.1.2 pIPFamily	250
8.206GetM2MAudioProfileReq Struct Reference	250
8.206.1 Detailed Description	250
8.206.2 Field Documentation	250
8.206.2.1 pGenerator	250
8.207GetM2MAudioProfileResp Struct Reference	250
8.207.1 Detailed Description	250
8.207.2 Field Documentation	251
8.207.2.1 CwtMute	251
8.207.2.2 EarMute	251
8.207.2.3 Generator	251
8.207.2.4 MicMute	251
8.207.2.5 Profile	251
8.207.2.6 Volume	251
8.208GetM2MAudioVolumeReq Struct Reference	251
8.208.1 Detailed Description	251
8.208.2 Field Documentation	252
8.208.2.1 Generator	252
8.208.2.2 Profile	252
8.209GetM2MAudioVolumeResp Struct Reference	252
8.209.1 Detailed Description	252

8.209.2 Field Documentation . . . . .	252
8.209.2.1 Level . . . . .	252
8.210GetM2MAVMuteReq Struct Reference . . . . .	252
8.210.1 Detailed Description . . . . .	252
8.210.2 Field Documentation . . . . .	252
8.210.2.1 Profile . . . . .	253
8.211GetM2MAVMuteResp Struct Reference . . . . .	253
8.211.1 Detailed Description . . . . .	253
8.211.2 Field Documentation . . . . .	253
8.211.2.1 CwtMute . . . . .	253
8.211.2.2 EarMute . . . . .	253
8.211.2.3 MicMute . . . . .	253
8.212GetM2MSpkrGainReq Struct Reference . . . . .	253
8.212.1 Detailed Description . . . . .	253
8.212.2 Field Documentation . . . . .	254
8.212.2.1 Profile . . . . .	254
8.213GetM2MSpkrGainResp Struct Reference . . . . .	254
8.213.1 Detailed Description . . . . .	254
8.213.2 Field Documentation . . . . .	254
8.213.2.1 Value . . . . .	254
8.214getMsgWaitingInfo Struct Reference . . . . .	254
8.214.1 Detailed Description . . . . .	254
8.214.2 Field Documentation . . . . .	255
8.214.2.1 msgWaitInfo . . . . .	255
8.214.2.2 numInstances . . . . .	255
8.215GetNetworkTimeResp Struct Reference . . . . .	255
8.215.1 Detailed Description . . . . .	255
8.215.2 Field Documentation . . . . .	255
8.215.2.1 p3GPP2TimeInfo . . . . .	255
8.215.2.2 p3GPPTimeInfo . . . . .	255
8.216GetRegMgrConfigParams Struct Reference . . . . .	255
8.216.1 Detailed Description . . . . .	255
8.216.2 Field Documentation . . . . .	256
8.216.2.1 pIMSTestMode . . . . .	256
8.216.2.2 pPCSCFPort . . . . .	256
8.216.2.3 pPriCSCFPortName . . . . .	256
8.216.2.4 pPriCSCFPortNameLen . . . . .	256
8.216.2.5 pSettingResp . . . . .	256
8.217GetSessionIDResp Struct Reference . . . . .	256
8.217.1 Field Documentation . . . . .	256

8.217.1.1 pSessionIDv4 . . . . .	256
8.217.1.2 pSessionIDv6 . . . . .	256
8.218GetSIPConfigResp Struct Reference . . . . .	256
8.218.1 Detailed Description . . . . .	257
8.218.2 Field Documentation . . . . .	257
8.218.2.1 pSettingResp . . . . .	257
8.218.2.2 pSigCompEnabled . . . . .	257
8.218.2.3 pSIPLocalPort . . . . .	257
8.218.2.4 pSubscribeTimer . . . . .	257
8.218.2.5 pTimerSIPReg . . . . .	257
8.218.2.6 pTimerT1 . . . . .	257
8.218.2.7 pTimerT2 . . . . .	257
8.218.2.8 pTimerTf . . . . .	257
8.219GnssData Struct Reference . . . . .	258
8.219.1 Detailed Description . . . . .	258
8.219.2 Field Documentation . . . . .	259
8.219.2.1 mask . . . . .	259
8.220gnssSvInfoNotification Struct Reference . . . . .	259
8.220.1 Detailed Description . . . . .	260
8.220.2 Field Documentation . . . . .	260
8.220.2.1 bAltitudeAssumed . . . . .	260
8.220.2.2 pSatelliteInfo . . . . .	260
8.221GPRSQoS Struct Reference . . . . .	260
8.221.1 Detailed Description . . . . .	260
8.221.2 Field Documentation . . . . .	261
8.221.2.1 delayClass . . . . .	261
8.221.2.2 meanThroughputClass . . . . .	261
8.221.2.3 peakThroughputClass . . . . .	261
8.221.2.4 precedenceClass . . . . .	261
8.221.2.5 reliabilityClass . . . . .	261
8.222GPRSRequestedQoS Struct Reference . . . . .	261
8.222.1 Detailed Description . . . . .	261
8.222.2 Field Documentation . . . . .	262
8.222.2.1 delayClass . . . . .	262
8.222.2.2 meanThroughputClass . . . . .	262
8.222.2.3 peakThroughputClass . . . . .	262
8.222.2.4 precedenceClass . . . . .	262
8.222.2.5 reliabilityClass . . . . .	262
8.223GPSSStateInfo Struct Reference . . . . .	262
8.223.1 Detailed Description . . . . .	262

8.223.2 Field Documentation	265
8.223.2.1 Altitude	265
8.223.2.2 EngineState	265
8.223.2.3 glo_almanac_sv_msk	265
8.223.2.4 glo_ephemeris_sv_msk	265
8.223.2.5 glo_health_sv_msk	265
8.223.2.6 glo_visible_sv_msk	265
8.223.2.7 gps_almanac_sv_msk	265
8.223.2.8 gps_ephemeris_sv_msk	265
8.223.2.9 gps_health_sv_msk	265
8.223.2.10gps_visible_sv_msk	265
8.223.2.11HorizontalUncertainty	265
8.223.2.12iono_valid	265
8.223.2.13Latitude	265
8.223.2.14Longitude	265
8.223.2.15sbas_almanac_sv_msk	265
8.223.2.16sbas_ephemeris_sv_msk	265
8.223.2.17sbas_health_sv_msk	266
8.223.2.18sbas_visible_sv_msk	266
8.223.2.19Time_uncert_ms	266
8.223.2.20TimeStmp_gps_week	266
8.223.2.21TimeStmp_tow_ms	266
8.223.2.22ValidMask	266
8.223.2.23VerticalUncertainty	266
8.223.2.24xtra_start_gps_minutes	266
8.223.2.25xtra_start_gps_week	266
8.223.2.26xtra_valid_duration_hours	266
8.224gpsTime_s Struct Reference	266
8.224.1 Detailed Description	266
8.224.2 Field Documentation	266
8.224.2.1 gpsTimeOfWeekMs	266
8.224.2.2 gpsWeek	266
8.225gsmCellInfo Struct Reference	266
8.225.1 Detailed Description	267
8.225.2 Field Documentation	267
8.225.2.1 arfcn	267
8.225.2.2 band1900	267
8.225.2.3 bsicld	267
8.225.2.4 cellldValid	267
8.225.2.5 rssi	267

8.225.2.6 srxlev . . . . .	267
8.226GSMRSSIThresh Struct Reference . . . . .	268
8.226.1 Detailed Description . . . . .	268
8.226.2 Field Documentation . . . . .	268
8.226.2.1 GSMRSSIThreshListLen . . . . .	268
8.226.2.2 pGSMRSSIThreshList . . . . .	268
8.227GSMSrvStatusInfo Struct Reference . . . . .	268
8.227.1 Detailed Description . . . . .	268
8.227.2 Field Documentation . . . . .	269
8.227.2.1 isPrefDataPath . . . . .	269
8.227.2.2 srvStatus . . . . .	269
8.227.2.3 trueSrvStatus . . . . .	269
8.228GSMSysInfo Struct Reference . . . . .	269
8.228.1 Detailed Description . . . . .	269
8.228.2 Field Documentation . . . . .	271
8.228.2.1 cellId . . . . .	271
8.228.2.2 cellIdValid . . . . .	271
8.228.2.3 dtmSupp . . . . .	271
8.228.2.4 dtmSuppValid . . . . .	271
8.228.2.5 egprsSupp . . . . .	271
8.228.2.6 egprsSuppValid . . . . .	271
8.228.2.7 lac . . . . .	272
8.228.2.8 lacValid . . . . .	272
8.228.2.9 MCC . . . . .	272
8.228.2.10MNC . . . . .	272
8.228.2.11networkIdValid . . . . .	272
8.228.2.12egRejectInfoValid . . . . .	272
8.228.2.13rejCause . . . . .	272
8.228.2.14rejectSrvDomain . . . . .	272
8.228.2.15sysInfoGSM . . . . .	272
8.229gyroAcceptReady_s Struct Reference . . . . .	272
8.229.1 Detailed Description . . . . .	272
8.229.2 Field Documentation . . . . .	272
8.229.2.1 batchPerSec . . . . .	272
8.229.2.2 injectEnable . . . . .	273
8.229.2.3 samplesPerBatch . . . . .	273
8.230gyroTempAcceptReady_s Struct Reference . . . . .	273
8.230.1 Detailed Description . . . . .	273
8.230.2 Field Documentation . . . . .	273
8.230.2.1 batchPerSec . . . . .	273

8.230.2.2 injectEnable . . . . .	273
8.230.2.3 samplesPerBatch . . . . .	273
8.231 HDRECIOThresh Struct Reference . . . . .	273
8.231.1 Detailed Description . . . . .	274
8.231.2 Field Documentation . . . . .	274
8.231.2.1 HDRECIOThreshListLen . . . . .	274
8.231.2.2 pHRECIOThreshList . . . . .	274
8.232 HDRIOTThresh Struct Reference . . . . .	274
8.232.1 Detailed Description . . . . .	274
8.232.2 Field Documentation . . . . .	274
8.232.2.1 HDRIOTThreshListLen . . . . .	274
8.232.2.2 pHDIOTThreshList . . . . .	274
8.233 HDRPersonalityInd Struct Reference . . . . .	274
8.233.1 Field Documentation . . . . .	275
8.233.1.1 pCurrentPersonality . . . . .	275
8.233.1.2 pPersonalityListLength . . . . .	275
8.233.1.3 pProtocolSubtypeElement . . . . .	275
8.234 HDRPersonalityResp Struct Reference . . . . .	275
8.234.1 Detailed Description . . . . .	275
8.234.2 Field Documentation . . . . .	275
8.234.2.1 pCurrentPersonality . . . . .	275
8.234.2.2 pPersonalityListLength . . . . .	275
8.234.2.3 pProtocolSubtypeElement . . . . .	275
8.235 HDRProtSubtypResp Struct Reference . . . . .	275
8.235.1 Detailed Description . . . . .	276
8.235.2 Field Documentation . . . . .	276
8.235.2.1 pAppSubType . . . . .	276
8.235.2.2 pCurrentPrsnlty . . . . .	276
8.235.2.3 pPersonalityListLength . . . . .	276
8.235.2.4 pProtoSubTypElmnt . . . . .	276
8.236 HDRRSSIThresh Struct Reference . . . . .	276
8.236.1 Detailed Description . . . . .	276
8.236.2 Field Documentation . . . . .	277
8.236.2.1 HDRRSSIThreshListLen . . . . .	277
8.236.2.2 pHDRRSSIThreshList . . . . .	277
8.237 HDRSINRThresh Struct Reference . . . . .	277
8.237.1 Detailed Description . . . . .	277
8.237.2 Field Documentation . . . . .	277
8.237.2.1 HDRSINRThresListLen . . . . .	277
8.237.2.2 pHDRSINRThresList . . . . .	277

8.238HDRSINRThreshold Struct Reference	277
8.238.1 Detailed Description	278
8.238.2 Field Documentation	278
8.238.2.1 HDRSINRThreshListLen	278
8.238.2.2 pHDRSINRThreshList	278
8.239HDRSSInfo Struct Reference	278
8.239.1 Detailed Description	278
8.239.2 Field Documentation	279
8.239.2.1 ecio	279
8.239.2.2 io	279
8.239.2.3 rssi	279
8.239.2.4 sinr	279
8.240hdrSSInfo Struct Reference	279
8.240.1 Detailed Description	279
8.240.2 Field Documentation	280
8.240.2.1 ecio	280
8.240.2.2 io	280
8.240.2.3 rssi	280
8.240.2.4 sinr	280
8.241HDRSysInfo Struct Reference	280
8.241.1 Detailed Description	280
8.241.2 Field Documentation	282
8.241.2.1 hdrActiveProt	282
8.241.2.2 hdrActiveProtValid	282
8.241.2.3 hdrPersonality	282
8.241.2.4 hdrPersonalityValid	282
8.241.2.5 is856SysId	282
8.241.2.6 is856SysIdValid	282
8.241.2.7 isSysPrIMatch	282
8.241.2.8 isSysPrIMatchValid	282
8.241.2.9 sysInfoHDR	282
8.242homeSIDNID Struct Reference	282
8.242.1 Detailed Description	282
8.242.2 Field Documentation	282
8.242.2.1 numInstances	282
8.242.2.2 SidNid	282
8.243hotSwapStatus Struct Reference	282
8.243.1 Detailed Description	283
8.243.2 Field Documentation	283
8.243.2.1 hotSwap	283

8.243.2.2 hotSwapLength . . . . .	283
8.244image_info_t Struct Reference . . . . .	283
8.244.1 Field Documentation . . . . .	283
8.244.1.1 buildID . . . . .	283
8.244.1.2 buildIDLen . . . . .	283
8.244.1.3 imageType . . . . .	283
8.244.1.4 uniqueID . . . . .	283
8.245ImageElement Struct Reference . . . . .	283
8.245.1 Detailed Description . . . . .	284
8.245.2 Field Documentation . . . . .	284
8.245.2.1 buildId . . . . .	284
8.245.2.2 buildIdLength . . . . .	284
8.245.2.3 imageId . . . . .	284
8.245.2.4 imageType . . . . .	284
8.246ImageIdElement Struct Reference . . . . .	284
8.246.1 Detailed Description . . . . .	284
8.246.2 Field Documentation . . . . .	285
8.246.2.1 buildID . . . . .	285
8.246.2.2 buildIDLength . . . . .	285
8.246.2.3 failureCount . . . . .	285
8.246.2.4 imageID . . . . .	285
8.246.2.5 storageIndex . . . . .	285
8.247ImageIDEntries Struct Reference . . . . .	285
8.247.1 Detailed Description . . . . .	285
8.247.2 Field Documentation . . . . .	286
8.247.2.1 executingImage . . . . .	286
8.247.2.2 imageIDElement . . . . .	286
8.247.2.3 imageIDSize . . . . .	286
8.247.2.4 imageType . . . . .	286
8.247.2.5 maxImages . . . . .	286
8.248ImageList Struct Reference . . . . .	286
8.248.1 Detailed Description . . . . .	286
8.248.2 Field Documentation . . . . .	286
8.248.2.1 imageIDEntries . . . . .	286
8.248.2.2 listSize . . . . .	286
8.249IMSAIndRegisterInfo Struct Reference . . . . .	286
8.249.1 Detailed Description . . . . .	287
8.249.2 Field Documentation . . . . .	287
8.249.2.1 pPdpStatusConfig . . . . .	287
8.249.2.2 pRatHandoverStatusConfig . . . . .	287

8.249.2.3 pRegStatusConfig . . . . .	287
8.249.2.4 pServiceStatusConfig . . . . .	287
8.250imsaPdpStatusInfo Struct Reference . . . . .	287
8.250.1 Detailed Description . . . . .	288
8.250.2 Field Documentation . . . . .	288
8.250.2.1 connetionState . . . . .	288
8.250.2.2 pFailErrorCode . . . . .	288
8.251imsaRatStatusInfo Struct Reference . . . . .	288
8.251.1 Detailed Description . . . . .	288
8.251.2 Field Documentation . . . . .	289
8.251.2.1 pErrorCodeStr . . . . .	289
8.251.2.2 pRATStatus . . . . .	289
8.251.2.3 pSrcRAT . . . . .	289
8.251.2.4 pTgtRAT . . . . .	289
8.252IMSARegistrationStatus Struct Reference . . . . .	289
8.252.1 Detailed Description . . . . .	289
8.252.2 Field Documentation . . . . .	290
8.252.2.1 plmsRegErrCode . . . . .	290
8.252.2.2 plmsRegStatus . . . . .	290
8.252.2.3 pNewlmsRegStatus . . . . .	290
8.253imsaRegStatusInfo Struct Reference . . . . .	290
8.253.1 Detailed Description . . . . .	290
8.253.2 Field Documentation . . . . .	290
8.253.2.1 pbIMSRegistered . . . . .	290
8.253.2.2 plmsRegStatus . . . . .	290
8.253.2.3 pRegStatusErrorCode . . . . .	290
8.254IMSAServiceStatus Struct Reference . . . . .	290
8.254.1 Detailed Description . . . . .	291
8.254.2 Field Documentation . . . . .	292
8.254.2.1 pSmsServiceRat . . . . .	292
8.254.2.2 pSmsServiceStatus . . . . .	292
8.254.2.3 pUtServiceRat . . . . .	292
8.254.2.4 pUtServiceStatus . . . . .	292
8.254.2.5 pVoipServiceRat . . . . .	292
8.254.2.6 pVoipServiceStatus . . . . .	292
8.254.2.7 pVsServiceRat . . . . .	292
8.254.2.8 pVsServiceStatus . . . . .	292
8.254.2.9 pVtServiceRat . . . . .	292
8.254.2.10pVtServiceStatus . . . . .	292
8.255IMSASupportedFieldsResp Struct Reference . . . . .	292

8.255.1 Detailed Description . . . . .	293
8.255.2 Field Documentation . . . . .	293
8.255.2.1 pIndFieldsList . . . . .	293
8.255.2.2 pReqFieldsList . . . . .	293
8.255.2.3 pRespFieldsList . . . . .	293
8.256IMSASupportedMsgInfo Struct Reference . . . . .	293
8.256.1 Detailed Description . . . . .	293
8.256.2 Field Documentation . . . . .	293
8.256.2.1 pSupportedMsgList . . . . .	293
8.257imsaSvcStatusInfo Struct Reference . . . . .	293
8.257.1 Detailed Description . . . . .	294
8.257.2 Field Documentation . . . . .	294
8.257.2.1 pSMSSvcRAT . . . . .	294
8.257.2.2 pSMSSvcStatus . . . . .	294
8.257.2.3 pUTSvcRAT . . . . .	294
8.257.2.4 pUTSvcStatus . . . . .	294
8.257.2.5 pVOIPSvcRAT . . . . .	294
8.257.2.6 pVOIPSvcStatus . . . . .	294
8.257.2.7 pVTSvcRAT . . . . .	294
8.257.2.8 pVTSvcStatus . . . . .	294
8.258imsCfgIndRegisterInfo Struct Reference . . . . .	294
8.258.1 Detailed Description . . . . .	295
8.258.2 Field Documentation . . . . .	295
8.258.2.1 pRegMgrConfigEvents . . . . .	295
8.258.2.2 pSIPConfigEvents . . . . .	295
8.258.2.3 pSMSConfigEvents . . . . .	295
8.258.2.4 pUserConfigEvents . . . . .	296
8.258.2.5 pVoIPConfigEvents . . . . .	296
8.259imsRegMgrConfigInfo Struct Reference . . . . .	296
8.259.1 Detailed Description . . . . .	296
8.259.2 Field Documentation . . . . .	296
8.259.2.1 pCSCFPortName . . . . .	296
8.259.2.2 pIMSTestMode . . . . .	296
8.259.2.3 pPriCSCFPort . . . . .	296
8.260imsSIPConfigInfo Struct Reference . . . . .	296
8.260.1 Detailed Description . . . . .	297
8.260.2 Field Documentation . . . . .	297
8.260.2.1 pSigCompEnabled . . . . .	297
8.260.2.2 pSIPLocalPort . . . . .	297
8.260.2.3 pSubscribeTimer . . . . .	297

8.260.2.4 pTimerSIPReg . . . . .	297
8.260.2.5 pTimerT1 . . . . .	297
8.260.2.6 pTimerT2 . . . . .	297
8.260.2.7 pTimerTf . . . . .	297
8.261imsSMSConfigInfo Struct Reference . . . . .	297
8.261.1 Detailed Description . . . . .	298
8.261.2 Field Documentation . . . . .	298
8.261.2.1 pPhoneCtxtURI . . . . .	298
8.261.2.2 pSMSFormat . . . . .	298
8.261.2.3 pSMSOverIPNwInd . . . . .	298
8.262imsUserConfigInfo Struct Reference . . . . .	298
8.262.1 Detailed Description . . . . .	298
8.262.2 Field Documentation . . . . .	298
8.262.2.1 pIMSDomain . . . . .	298
8.263imsVoIPConfigInfo Struct Reference . . . . .	299
8.263.1 Detailed Description . . . . .	299
8.263.2 Field Documentation . . . . .	301
8.263.2.1 pAmrMode . . . . .	301
8.263.2.2 pAmrOctetAligned . . . . .	301
8.263.2.3 pAmrWbEnable . . . . .	301
8.263.2.4 pAmrWBMode . . . . .	301
8.263.2.5 pAmrWBOctetAligned . . . . .	301
8.263.2.6 pMinSessionExpiryTimer . . . . .	301
8.263.2.7 pRingBackTimer . . . . .	301
8.263.2.8 pRingingTimer . . . . .	301
8.263.2.9 pRTPRTCPInactTimer . . . . .	301
8.263.2.10pScrAmrEnable . . . . .	301
8.263.2.11pScrAmrWbEnable . . . . .	301
8.263.2.12pSessionExpiryTimer . . . . .	301
8.264IndFieldsList Struct Reference . . . . .	301
8.264.1 Detailed Description . . . . .	301
8.264.2 Field Documentation . . . . .	301
8.264.2.1 indicationFields . . . . .	301
8.264.2.2 indicationFieldsLen . . . . .	301
8.265infoInterFreq Struct Reference . . . . .	302
8.265.1 Detailed Description . . . . .	302
8.265.2 Field Documentation . . . . .	302
8.265.2.1 cell_resel_priority . . . . .	302
8.265.2.2 cellInterFreqParams . . . . .	302
8.265.2.3 cells_len . . . . .	302

8.265.2.4 earfcn . . . . .	303
8.265.2.5 threshXHigh . . . . .	303
8.265.2.6 threshXLow . . . . .	303
8.266IOThresh Struct Reference . . . . .	303
8.266.1 Detailed Description . . . . .	303
8.266.2 Field Documentation . . . . .	303
8.266.2.1 IOThresListLen . . . . .	303
8.266.2.2 pIOThresList . . . . .	303
8.267IPv4Addr Struct Reference . . . . .	303
8.267.1 Detailed Description . . . . .	303
8.267.2 Field Documentation . . . . .	304
8.267.2.1 addr . . . . .	304
8.267.2.2 subnetMask . . . . .	304
8.268IPv6Addr Struct Reference . . . . .	304
8.268.1 Detailed Description . . . . .	304
8.268.2 Field Documentation . . . . .	304
8.268.2.1 addr . . . . .	304
8.268.2.2 prefixLen . . . . .	304
8.269IPv6AddressInfo Struct Reference . . . . .	304
8.269.1 Detailed Description . . . . .	304
8.269.2 Field Documentation . . . . .	305
8.269.2.1 IPAddressV6 . . . . .	305
8.269.2.2 IPV6PrefixLen . . . . .	305
8.270ipv6AddressInfo Struct Reference . . . . .	305
8.270.1 Detailed Description . . . . .	305
8.270.2 Field Documentation . . . . .	305
8.270.2.1 IPAddressV6 . . . . .	305
8.270.2.2 IPV6PrefixLen . . . . .	305
8.271IPv6GWAddressInfo Struct Reference . . . . .	305
8.271.1 Detailed Description . . . . .	305
8.271.2 Field Documentation . . . . .	306
8.271.2.1 gwAddressV6 . . . . .	306
8.271.2.2 gwV6PrefixLen . . . . .	306
8.272IPv6TrafCls Struct Reference . . . . .	306
8.272.1 Detailed Description . . . . .	306
8.272.2 Field Documentation . . . . .	306
8.272.2.1 mask . . . . .	306
8.272.2.2 val . . . . .	306
8.273LibPackGPRSRequestedQoS Struct Reference . . . . .	306
8.273.1 Detailed Description . . . . .	307

8.273.2 Field Documentation	307
8.273.2.1 delayClass	307
8.273.2.2 meanThroughputClass	307
8.273.2.3 peakThroughputClass	307
8.273.2.4 precedenceClass	307
8.273.2.5 reliabilityClass	307
8.274 LibpackProfile3GPP Struct Reference	307
8.274.1 Detailed Description	308
8.274.2 Field Documentation	312
8.274.2.1 pAddrAllocPref	312
8.274.2.2 pAPNClass	312
8.274.2.3 pAPNDisabledFlag	312
8.274.2.4 pAPNName	312
8.274.2.5 pAPNnameSize	312
8.274.2.6 pAuthenticationPref	312
8.274.2.7 pGPRSMinimumQoS	312
8.274.2.8 pGPRSRequestedQos	312
8.274.2.9 plmCnFlag	312
8.274.2.10 pIPv4AddrPref	312
8.274.2.11 pIPv6AddPref	312
8.274.2.12 pPassword	312
8.274.2.13 pPasswordSize	312
8.274.2.14 pPcscfAddrUsingDhcp	312
8.274.2.15 pPcscfAddrUsingPCO	312
8.274.2.16 pPDNInactivTimeout	312
8.274.2.17 pPdpAccessConFlag	312
8.274.2.18 pPdpContext	312
8.274.2.19 pPdpDataCompType	312
8.274.2.20 pPdpHdrCompType	312
8.274.2.21 pPDPTtype	312
8.274.2.22 pPriDNSIPv4AddPref	312
8.274.2.23 pPriDNSIPv6addpref	312
8.274.2.24 pPrimaryID	313
8.274.2.25 pProfilename	313
8.274.2.26 pProfilenameSize	313
8.274.2.27 pQosClassID	313
8.274.2.28 pSecDNSIPv4AddPref	313
8.274.2.29 pSecDNSIPv6addpref	313
8.274.2.30 pSecondaryFlag	313
8.274.2.31 pTFTID1Params	313

8.274.2.32	pTFTID2Params	313
8.274.2.33	pUMTSMInQoS	313
8.274.2.34	pUMTSMInQoSSigInd	313
8.274.2.35	pUMTSReqQoS	313
8.274.2.36	pUMTSReqQoSSigInd	313
8.274.2.37	pUsername	313
8.274.2.38	pUsernameSize	313
8.275	LibpackProfile3GPP2 Struct Reference	313
8.275.1	Detailed Description	314
8.275.2	Field Documentation	317
8.275.2.1	pAllowLinger	317
8.275.2.2	pAPNClass3GPP2	317
8.275.2.3	pAPNEnabled3GPP2	317
8.275.2.4	pApnString	317
8.275.2.5	pApnStringSize	317
8.275.2.6	pAppPriority	317
8.275.2.7	pAppType	317
8.275.2.8	pAuthPassword	317
8.275.2.9	pAuthPasswordSize	317
8.275.2.10	pAuthProtocol	318
8.275.2.11	pAuthRetryCount	318
8.275.2.12	pAuthTimeout	318
8.275.2.13	pDataMode	318
8.275.2.14	pDataRate	318
8.275.2.15	pIpcpAckTimeout	318
8.275.2.16	pIpcpCreqRetryCount	318
8.275.2.17	pIsPcscfAddressNedded	318
8.275.2.18	pLcpAckTimeout	318
8.275.2.19	pLcpCreqRetryCount	318
8.275.2.20	pNegoDnsSrvrPref	318
8.275.2.21	pPDNInactivTimeout3GPP2	318
8.275.2.22	pPdnType	318
8.275.2.23	pPppSessCloseTimer1x	318
8.275.2.24	pPppSessCloseTimerDO	318
8.275.2.25	pPrimaryV4DnsAddress	318
8.275.2.26	pPriV6DnsAddress	318
8.275.2.27	pRATType	318
8.275.2.28	pSecondaryV4DnsAddress	318
8.275.2.29	pSecV6DnsAddress	318
8.275.2.30	pUserId	318

8.275.2.31p	UserIdSize	318
8.276	LibPackprofile_3GPP Struct Reference	318
8.276.1	Detailed Description	319
8.276.2	Field Documentation	323
8.276.2.1	pAddrAllocPref	323
8.276.2.2	pAPNClass	323
8.276.2.3	pAPNDisabledFlag	323
8.276.2.4	pAPNName	323
8.276.2.5	pAPNnameSize	323
8.276.2.6	pAuthenticationPref	323
8.276.2.7	pGPRSMinimumQoS	323
8.276.2.8	pGPRSRequestedQos	323
8.276.2.9	plmCnFlag	323
8.276.2.10	pIPv4AddrPref	323
8.276.2.11	pIPv6AddPref	323
8.276.2.12	pPassword	323
8.276.2.13	pPasswordSize	323
8.276.2.14	pPcscfAddrUsingDhcp	323
8.276.2.15	pPcscfAddrUsingPCO	323
8.276.2.16	pPDNInactivTimeout	323
8.276.2.17	pPdpAccessConFlag	323
8.276.2.18	pPdpContext	323
8.276.2.19	pPdpDataCompType	323
8.276.2.20	pPdpHdrCompType	323
8.276.2.21	pPDType	323
8.276.2.22	pPriDNSIPv4AddPref	323
8.276.2.23	pPriDNSIPv6addpref	323
8.276.2.24	pPrimaryID	324
8.276.2.25	pProfilename	324
8.276.2.26	pProfilenameSize	324
8.276.2.27	pQosClassID	324
8.276.2.28	pSecDNSIPv4AddPref	324
8.276.2.29	pSecDNSIPv6addpref	324
8.276.2.30	pSecondaryFlag	324
8.276.2.31	pTFTID1Params	324
8.276.2.32	pTFTID2Params	324
8.276.2.33	pUMTSMinQoS	324
8.276.2.34	pUMTSMinQoSSigInd	324
8.276.2.35	pUMTSReqQoS	324
8.276.2.36	pUMTSReqQoSSigInd	324

8.276.2.37pUsername . . . . .	324
8.276.2.38pUsernameSize . . . . .	324
8.277LibPackprofile_3GPP2 Struct Reference . . . . .	324
8.277.1 Detailed Description . . . . .	325
8.277.2 Field Documentation . . . . .	328
8.277.2.1 pAllowLinger . . . . .	328
8.277.2.2 pAPNClass3GPP2 . . . . .	328
8.277.2.3 pAPNEnabled3GPP2 . . . . .	328
8.277.2.4 pApnString . . . . .	328
8.277.2.5 pApnStringSize . . . . .	328
8.277.2.6 pAppPriority . . . . .	328
8.277.2.7 pAppType . . . . .	328
8.277.2.8 pAuthPassword . . . . .	328
8.277.2.9 pAuthPassword_tSize . . . . .	328
8.277.2.10pAuthProtocol . . . . .	329
8.277.2.11pAuthRetryCount . . . . .	329
8.277.2.12pAuthTimeout . . . . .	329
8.277.2.13pDataMode . . . . .	329
8.277.2.14pDataRate . . . . .	329
8.277.2.15pIpcpAckTimeout . . . . .	329
8.277.2.16pIpcpCreqRetryCount . . . . .	329
8.277.2.17pIsPcscfAddressNedded . . . . .	329
8.277.2.18pLcpAckTimeout . . . . .	329
8.277.2.19pLcpCreqRetryCount . . . . .	329
8.277.2.20pNegoDnsSrvrPref . . . . .	329
8.277.2.21pPDNInactivTimeout3GPP2 . . . . .	329
8.277.2.22pPdnType . . . . .	329
8.277.2.23pPppSessCloseTimer1x . . . . .	329
8.277.2.24pPppSessCloseTimerDO . . . . .	329
8.277.2.25pPrimaryV4DnsAddress . . . . .	329
8.277.2.26pPriV6DnsAddress . . . . .	329
8.277.2.27pRATType . . . . .	329
8.277.2.28pSecondaryV4DnsAddress . . . . .	329
8.277.2.29pSecV6DnsAddress . . . . .	329
8.277.2.30pUserId . . . . .	329
8.277.2.31pUserIdSize . . . . .	329
8.278LibPackQosClassID Struct Reference . . . . .	329
8.278.1 Detailed Description . . . . .	330
8.278.2 Field Documentation . . . . .	330
8.278.2.1 gDIBitRate . . . . .	330

8.278.2.2 gUIBitRate . . . . .	330
8.278.2.3 maxDIBitRate . . . . .	330
8.278.2.4 maxUIBitRate . . . . .	330
8.278.2.5 QCI . . . . .	330
8.279LibPackTFTIDParams Struct Reference . . . . .	330
8.279.1 Detailed Description . . . . .	331
8.279.2 Field Documentation . . . . .	331
8.279.2.1 destPortRangeEnd . . . . .	332
8.279.2.2 destPortRangeStart . . . . .	332
8.279.2.3 eValid . . . . .	332
8.279.2.4 filterId . . . . .	332
8.279.2.5 flowLabel . . . . .	332
8.279.2.6 IPSECSPi . . . . .	332
8.279.2.7 ipVersion . . . . .	332
8.279.2.8 nextHeader . . . . .	332
8.279.2.9 pSourceIP . . . . .	332
8.279.2.10sourceIPMask . . . . .	332
8.279.2.11srcPortRangeEnd . . . . .	332
8.279.2.12srcPortRangeStart . . . . .	332
8.279.2.13osMask . . . . .	332
8.280LibPackUMTSQoS Struct Reference . . . . .	332
8.280.1 Detailed Description . . . . .	332
8.280.2 Field Documentation . . . . .	334
8.280.2.1 deliveryErrSDU . . . . .	334
8.280.2.2 grntDownlinkBitrate . . . . .	334
8.280.2.3 grntUplinkBitrate . . . . .	334
8.280.2.4 maxDownlinkBitrate . . . . .	334
8.280.2.5 maxSDUSize . . . . .	334
8.280.2.6 maxUplinkBitrate . . . . .	334
8.280.2.7 qosDeliveryOrder . . . . .	334
8.280.2.8 resBerRatio . . . . .	334
8.280.2.9 sduErrorRatio . . . . .	334
8.280.2.10trafficClass . . . . .	334
8.280.2.11trafficPriority . . . . .	334
8.280.2.12transferDelay . . . . .	334
8.281LibPackUMTSReqQoSsigInd Struct Reference . . . . .	334
8.281.1 Detailed Description . . . . .	335
8.281.2 Field Documentation . . . . .	335
8.281.2.1 SigInd . . . . .	335
8.281.2.2 UMTSReqQoS . . . . .	335

8.282lineCtrlInfo Struct Reference	335
8.282.1 Detailed Description	335
8.282.2 Field Documentation	336
8.282.2.1 polarityIncluded	336
8.282.2.2 pwrDenialTime	336
8.282.2.3 revPolarity	336
8.282.2.4 toggleMode	336
8.283loc_BdsSV Struct Reference	336
8.283.1 Detailed Description	336
8.283.2 Field Documentation	336
8.283.2.1 id	336
8.283.2.2 mask	336
8.284loc_BdsSVInfo Struct Reference	336
8.284.1 Detailed Description	336
8.284.2 Field Documentation	337
8.284.2.1 len	337
8.284.2.2 pSV	337
8.285loc_CellDb Struct Reference	337
8.285.1 Detailed Description	337
8.285.2 Field Documentation	337
8.285.2.1 mask	337
8.286loc_ClkInfo Struct Reference	337
8.286.1 Detailed Description	338
8.286.2 Field Documentation	338
8.286.2.1 mask	339
8.287loc_GnssData Struct Reference	339
8.287.1 Detailed Description	339
8.287.2 Field Documentation	340
8.287.2.1 mask	340
8.288loc_gpsTime Struct Reference	340
8.288.1 Detailed Description	341
8.288.2 Field Documentation	341
8.288.2.1 gpsTimeOfWeekMs	341
8.288.2.2 gpsWeek	341
8.289loc_LocApplicationInfo Struct Reference	341
8.289.1 Detailed Description	341
8.289.2 Field Documentation	342
8.289.2.1 appNameLength	342
8.289.2.2 appProviderLength	342
8.289.2.3 appVersionLength	342

8.289.2.4 appVersionValid . . . . .	342
8.289.2.5 pAppName . . . . .	342
8.289.2.6 pAppProvider . . . . .	342
8.289.2.7 pAppVersion . . . . .	342
8.290loc_precisionDilution Struct Reference . . . . .	342
8.290.1 Detailed Description . . . . .	342
8.290.2 Field Documentation . . . . .	343
8.290.2.1 HDOP . . . . .	343
8.290.2.2 PDOP . . . . .	343
8.290.2.3 VDOP . . . . .	343
8.291loc_sensorDataUsage Struct Reference . . . . .	343
8.291.1 Detailed Description . . . . .	343
8.291.2 Field Documentation . . . . .	343
8.291.2.1 aidingIndicatorMask . . . . .	344
8.291.2.2 usageMask . . . . .	344
8.292loc_SV Struct Reference . . . . .	344
8.292.1 Detailed Description . . . . .	344
8.292.2 Field Documentation . . . . .	344
8.292.2.1 id . . . . .	344
8.292.2.2 mask . . . . .	344
8.292.2.3 system . . . . .	344
8.293loc_SVInfo Struct Reference . . . . .	344
8.293.1 Detailed Description . . . . .	345
8.293.2 Field Documentation . . . . .	345
8.293.2.1 len . . . . .	345
8.293.2.2 pSV . . . . .	345
8.294loc_svUsedforFix Struct Reference . . . . .	345
8.294.1 Detailed Description . . . . .	345
8.294.2 Field Documentation . . . . .	346
8.294.2.1 gnssSvUsedList . . . . .	346
8.294.2.2 gnssSvUsedList_len . . . . .	346
8.295LocApplicationInfo Struct Reference . . . . .	346
8.295.1 Detailed Description . . . . .	346
8.295.2 Field Documentation . . . . .	346
8.295.2.1 appNameLength . . . . .	347
8.295.2.2 appProviderLength . . . . .	347
8.295.2.3 appVersionLength . . . . .	347
8.295.2.4 appVersionValid . . . . .	347
8.295.2.5 pAppName . . . . .	347
8.295.2.6 pAppProvider . . . . .	347

8.295.2.7 pAppVersion . . . . .	347
8.296LocDelAssDataReq Struct Reference . . . . .	347
8.296.1 Detailed Description . . . . .	347
8.296.2 Field Documentation . . . . .	347
8.296.2.1 pBdsSVInfo . . . . .	348
8.296.2.2 pCellDb . . . . .	348
8.296.2.3 pClkInfo . . . . .	348
8.296.2.4 pGnssData . . . . .	348
8.296.2.5 pSVInfo . . . . .	348
8.297LOCEventRegisterReqResp Struct Reference . . . . .	348
8.297.1 Detailed Description . . . . .	348
8.297.2 Field Documentation . . . . .	350
8.297.2.1 eventRegister . . . . .	350
8.298LOCExtPowerStateReqResp Struct Reference . . . . .	350
8.298.1 Detailed Description . . . . .	350
8.298.2 Field Documentation . . . . .	350
8.298.2.1 extPowerState . . . . .	350
8.299LocInjectPositionReq Struct Reference . . . . .	350
8.299.1 Detailed Description . . . . .	351
8.299.2 Field Documentation . . . . .	353
8.299.2.1 pAltitudeSrcInfo . . . . .	353
8.299.2.2 pAltitudeWrtEllipsoid . . . . .	353
8.299.2.3 pAltitudeWrtMeanSeaLevel . . . . .	353
8.299.2.4 pHorConfidence . . . . .	354
8.299.2.5 pHorReliability . . . . .	354
8.299.2.6 pHorUncCircular . . . . .	354
8.299.2.7 pLatitude . . . . .	354
8.299.2.8 pLongitude . . . . .	354
8.299.2.9 pPositionSrc . . . . .	354
8.299.2.10pRawHorConfidence . . . . .	354
8.299.2.11pRawHorUncCircular . . . . .	354
8.299.2.12pTimestampAge . . . . .	354
8.299.2.13pTimestampUtc . . . . .	354
8.299.2.14pVertConfidence . . . . .	354
8.299.2.15pVertReliability . . . . .	354
8.299.2.16pVertUnc . . . . .	354
8.300LocInjectSensorDataReq Struct Reference . . . . .	354
8.300.1 Detailed Description . . . . .	354
8.300.2 Field Documentation . . . . .	355
8.300.2.1 pAcceleroData . . . . .	355

8.300.2.2 pAcceleroTempData . . . . .	355
8.300.2.3 pAcceleroTimeSrc . . . . .	355
8.300.2.4 pGyroData . . . . .	355
8.300.2.5 pGyroTempData . . . . .	355
8.300.2.6 pGyroTimeSrc . . . . .	355
8.300.2.7 pOpaqueIdentifier . . . . .	355
8.301 LocSetCradleMountReq Struct Reference . . . . .	355
8.301.1 Detailed Description . . . . .	356
8.301.2 Field Documentation . . . . .	356
8.301.2.1 pConfidence . . . . .	356
8.301.2.2 state . . . . .	356
8.302 LOCStartReq Struct Reference . . . . .	356
8.302.1 Detailed Description . . . . .	356
8.302.2 Field Documentation . . . . .	357
8.302.2.1 pApplicationInfo . . . . .	357
8.302.2.2 pConfigAltitudeAssumed . . . . .	357
8.302.2.3 pHorizontalAccuracyLvl . . . . .	358
8.302.2.4 pIntermediateReportState . . . . .	358
8.302.2.5 pMinIntervalTime . . . . .	358
8.302.2.6 pRecurrenceType . . . . .	358
8.302.2.7 SessionId . . . . .	358
8.303 LOCStopReq Struct Reference . . . . .	358
8.303.1 Detailed Description . . . . .	358
8.303.2 Field Documentation . . . . .	358
8.303.2.1 sessionId . . . . .	358
8.304 LteCQIParm Struct Reference . . . . .	358
8.304.1 Detailed Description . . . . .	358
8.304.2 Field Documentation . . . . .	359
8.304.2.1 CQIValueCW0 . . . . .	359
8.304.2.2 CQIValueCW1 . . . . .	359
8.304.2.3 ValidityCW0 . . . . .	359
8.304.2.4 ValidityCW1 . . . . .	359
8.305 lteEARFCN Struct Reference . . . . .	359
8.305.1 Detailed Description . . . . .	359
8.305.2 Field Documentation . . . . .	359
8.305.2.1 earfcn0 . . . . .	360
8.305.2.2 earfcn1 . . . . .	360
8.305.2.3 status . . . . .	360
8.306 lteGsmCellInfo Struct Reference . . . . .	360
8.306.1 Detailed Description . . . . .	360

8.306.2 Field Documentation	361
8.306.2.1 cellReselPriority	361
8.306.2.2 cells_len	361
8.306.2.3 GsmCellInfo	361
8.306.2.4 nccPermitted	361
8.306.2.5 threshGsmHigh	361
8.306.2.6 threshGsmLow	361
8.307LTEInfo Struct Reference	361
8.307.1 Detailed Description	361
8.307.2 Field Documentation	363
8.307.2.1 band	363
8.307.2.2 bandwidth	363
8.307.2.3 emmConnState	363
8.307.2.4 emmState	363
8.307.2.5 emmSubState	363
8.307.2.6 RXChan	363
8.307.2.7 TXChan	363
8.308LTEInfoInterfreq Struct Reference	363
8.308.1 Detailed Description	363
8.308.2 Field Documentation	363
8.308.2.1 freqsLen	363
8.308.2.2 InfoInterfreq	363
8.308.2.3 ueInIdle	363
8.309LTEInfoIntrafreq Struct Reference	363
8.309.1 Detailed Description	364
8.309.2 Field Documentation	365
8.309.2.1 CellParams	365
8.309.2.2 cellReselPriority	365
8.309.2.3 cellsLen	365
8.309.2.4 earfcn	365
8.309.2.5 globalCellId	365
8.309.2.6 plmn	365
8.309.2.7 servingCellId	365
8.309.2.8 sIntraSearch	365
8.309.2.9 sNonIntraSearch	365
8.309.2.10tac	365
8.309.2.11threshServingLow	365
8.309.2.12ueInIdle	366
8.310LTEInfoNeighboringGSM Struct Reference	366
8.310.1 Detailed Description	366

8.310.2 Field Documentation . . . . .	366
8.310.2.1 freqsLen . . . . .	366
8.310.2.2 LteGsmCellInfo . . . . .	366
8.310.2.3 ueInIdle . . . . .	366
8.311 LTEInfoNeighboringWCDMA Struct Reference . . . . .	366
8.311.1 Detailed Description . . . . .	366
8.311.2 Field Documentation . . . . .	367
8.311.2.1 freqsLen . . . . .	367
8.311.2.2 LTEWCDMACellInfo . . . . .	367
8.311.2.3 ueInIdle . . . . .	367
8.312 LteNasReleaseInfo_s Struct Reference . . . . .	367
8.312.1 Detailed Description . . . . .	367
8.312.2 Field Documentation . . . . .	367
8.312.2.1 nas_major . . . . .	367
8.312.2.2 nas_minor . . . . .	367
8.312.2.3 nas_release . . . . .	368
8.313 ItePCI Struct Reference . . . . .	368
8.313.1 Detailed Description . . . . .	368
8.313.2 Field Documentation . . . . .	368
8.313.2.1 earfcn . . . . .	368
8.313.2.2 pci . . . . .	368
8.313.2.3 status . . . . .	368
8.314 IteRsrpInformation Struct Reference . . . . .	368
8.314.1 Detailed Description . . . . .	368
8.314.2 Field Documentation . . . . .	369
8.314.2.1 rsrplevel . . . . .	369
8.315 LTERSRPThresh Struct Reference . . . . .	369
8.315.1 Detailed Description . . . . .	369
8.315.2 Field Documentation . . . . .	369
8.315.2.1 LTERSRPThreshListLen . . . . .	369
8.315.2.2 pLTERSRPThreshList . . . . .	369
8.316 LTERSRQThresh Struct Reference . . . . .	369
8.316.1 Detailed Description . . . . .	369
8.316.2 Field Documentation . . . . .	370
8.316.2.1 LTERSRQThreshListLen . . . . .	370
8.316.2.2 pLTERSRQThreshList . . . . .	370
8.317 LTERSSIThresh Struct Reference . . . . .	370
8.317.1 Detailed Description . . . . .	370
8.317.2 Field Documentation . . . . .	370
8.317.2.1 LTERSSIThreshListLen . . . . .	370

8.317.2.2 pLTERSSIThreshList . . . . .	370
8.318LTESigRptCfg Struct Reference . . . . .	370
8.318.1 Detailed Description . . . . .	370
8.318.2 Field Documentation . . . . .	371
8.318.2.1 avgPeriod . . . . .	371
8.318.2.2 rptRate . . . . .	371
8.319LTESigRptConfig Struct Reference . . . . .	371
8.319.1 Detailed Description . . . . .	371
8.319.2 Field Documentation . . . . .	372
8.319.2.1 avgPeriod . . . . .	372
8.319.2.2 rptRate . . . . .	372
8.320lteSnrinformation Struct Reference . . . . .	372
8.320.1 Detailed Description . . . . .	372
8.320.2 Field Documentation . . . . .	372
8.320.2.1 snrlevel . . . . .	373
8.321LTESNRThresh Struct Reference . . . . .	373
8.321.1 Detailed Description . . . . .	373
8.321.2 Field Documentation . . . . .	373
8.321.2.1 LTESNRThresListLen . . . . .	373
8.321.2.2 pLTESNRThresList . . . . .	373
8.322LTESNRThreshold Struct Reference . . . . .	373
8.322.1 Detailed Description . . . . .	373
8.322.2 Field Documentation . . . . .	374
8.322.2.1 LTESNRThreshListLen . . . . .	374
8.322.2.2 pLTESNRThreshList . . . . .	374
8.323LTESInfo Struct Reference . . . . .	374
8.323.1 Detailed Description . . . . .	374
8.323.2 Field Documentation . . . . .	374
8.323.2.1 rsrp . . . . .	374
8.323.2.2 rsrq . . . . .	374
8.323.2.3 rssi . . . . .	375
8.323.2.4 snr . . . . .	375
8.324lteSSInfo Struct Reference . . . . .	375
8.324.1 Detailed Description . . . . .	375
8.324.2 Field Documentation . . . . .	375
8.324.2.1 rsrp . . . . .	375
8.324.2.2 rsrq . . . . .	375
8.324.2.3 rssi . . . . .	375
8.324.2.4 snr . . . . .	375
8.325LTESysInfo Struct Reference . . . . .	375

8.325.1 Detailed Description	375
8.325.2 Field Documentation	377
8.325.2.1 cellId	377
8.325.2.2 cellIdValid	377
8.325.2.3 lac	377
8.325.2.4 lacValid	377
8.325.2.5 MCC	377
8.325.2.6 MNC	377
8.325.2.7 networkIdValid	377
8.325.2.8 regRejectInfoValid	377
8.325.2.9 rejCause	377
8.325.2.10 rejectSrvDomain	377
8.325.2.11 sysInfoLTE	377
8.325.2.12 tac	377
8.325.2.13 tacValid	377
8.326 lteWcdmaCellInfo Struct Reference	377
8.326.1 Detailed Description	378
8.326.2 Field Documentation	378
8.326.2.1 cellReselPriority	378
8.326.2.2 cellsLen	378
8.326.2.3 threshXhigh	378
8.326.2.4 threshXlow	378
8.326.2.5 uarfcn	378
8.326.2.6 WCDMACellInfo	379
8.327 messageModeTlv Struct Reference	379
8.327.1 Detailed Description	379
8.327.2 Field Documentation	379
8.327.2.1 MessageModeInfo	379
8.327.2.2 TlvPresent	379
8.328 messageWaitingInfoContent Struct Reference	379
8.328.1 Detailed Description	379
8.328.2 Field Documentation	380
8.328.2.1 activeInd	380
8.328.2.2 msgCount	380
8.328.2.3 msgType	380
8.329 minBasedIMSI Struct Reference	380
8.329.1 Detailed Description	380
8.329.2 Field Documentation	380
8.329.2.1 imsiM1112	380
8.329.2.2 imsiMS1	380

8.329.2.3 imsiMS2 . . . . .	381
8.329.2.4 mccM . . . . .	381
8.330mitigationDevList Struct Reference . . . . .	381
8.330.1 Detailed Description . . . . .	381
8.330.2 Field Documentation . . . . .	381
8.330.2.1 maxMitigationLevel . . . . .	381
8.330.2.2 mitigationDevId . . . . .	381
8.330.2.3 mitigationDevIdLen . . . . .	381
8.331MNRInfo Struct Reference . . . . .	381
8.331.1 Detailed Description . . . . .	381
8.331.2 Field Documentation . . . . .	382
8.331.2.1 mcc . . . . .	382
8.331.2.2 mnc . . . . .	382
8.331.2.3 rat . . . . .	382
8.332ModifyProfileIn Struct Reference . . . . .	382
8.332.1 Detailed Description . . . . .	382
8.332.2 Field Documentation . . . . .	383
8.332.2.1 curProfile . . . . .	383
8.332.2.2 pProfileID . . . . .	383
8.332.2.3 pProfileType . . . . .	383
8.333ModifyProfileOut Struct Reference . . . . .	383
8.333.1 Detailed Description . . . . .	383
8.333.2 Field Documentation . . . . .	383
8.333.2.1 pExtErrorCode . . . . .	383
8.334msgWaitingInfo Struct Reference . . . . .	383
8.334.1 Detailed Description . . . . .	383
8.334.2 Field Documentation . . . . .	384
8.334.2.1 msgWaitInfo . . . . .	384
8.334.2.2 numInstances . . . . .	384
8.335namName Struct Reference . . . . .	384
8.335.1 Detailed Description . . . . .	384
8.335.2 Field Documentation . . . . .	384
8.335.2.1 namName . . . . .	384
8.335.2.2 namNameLen . . . . .	384
8.336nas_acqOrderPref Struct Reference . . . . .	384
8.336.1 Detailed Description . . . . .	384
8.336.2 Field Documentation . . . . .	385
8.336.2.1 acqOrdeLen . . . . .	385
8.336.2.2 pAcqOrder . . . . .	385
8.337nas_AddCDMASysInfo Struct Reference . . . . .	385

8.337.1 Detailed Description . . . . .	385
8.337.2 Field Documentation . . . . .	385
8.337.2.1 geoSysIdx . . . . .	385
8.337.2.2 regPrd . . . . .	385
8.338nas_AddSysInfo Struct Reference . . . . .	385
8.338.1 Detailed Description . . . . .	386
8.338.2 Field Documentation . . . . .	386
8.338.2.1 cellBroadcastCap . . . . .	386
8.338.2.2 geoSysIdx . . . . .	386
8.339nas_CallBarringSysInfo Struct Reference . . . . .	386
8.339.1 Detailed Description . . . . .	386
8.339.2 Field Documentation . . . . .	387
8.339.2.1 csBarStatus . . . . .	387
8.339.2.2 psBarStatus . . . . .	387
8.340nas_callBarStatus Struct Reference . . . . .	387
8.340.1 Detailed Description . . . . .	387
8.340.2 Field Documentation . . . . .	388
8.340.2.1 csBarStatus . . . . .	388
8.340.2.2 psBarStatus . . . . .	388
8.341nas_CDMAECIOThresh Struct Reference . . . . .	388
8.341.1 Detailed Description . . . . .	388
8.341.2 Field Documentation . . . . .	388
8.341.2.1 CDMAECIOThreshListLen . . . . .	388
8.341.2.2 pCDMAECIOThreshList . . . . .	388
8.342nas_CDMAInfo Struct Reference . . . . .	388
8.342.1 Detailed Description . . . . .	388
8.342.2 Field Documentation . . . . .	389
8.342.2.1 baseId . . . . .	389
8.342.2.2 baseLat . . . . .	389
8.342.2.3 baseLong . . . . .	389
8.342.2.4 nid . . . . .	389
8.342.2.5 refpn . . . . .	389
8.342.2.6 sid . . . . .	389
8.343nas_CDMARSSIThresh Struct Reference . . . . .	389
8.343.1 Detailed Description . . . . .	389
8.343.2 Field Documentation . . . . .	390
8.343.2.1 CDMARSSIThreshListLen . . . . .	390
8.343.2.2 pCDMARSSIThreshList . . . . .	390
8.344nas_CDMASysInfo Struct Reference . . . . .	390
8.344.1 Detailed Description . . . . .	390

8.344.2 Field Documentation	393
8.344.2.1 baseId	393
8.344.2.2 baseLat	393
8.344.2.3 baseLong	393
8.344.2.4 bsInfoValid	393
8.344.2.5 bsPRev	393
8.344.2.6 bsPRevValid	393
8.344.2.7 ccsSupported	393
8.344.2.8 ccsSupportedValid	393
8.344.2.9 cdmaSysIdValid	393
8.344.2.10 isSysPrIMatch	393
8.344.2.11 isSysPrIMatchValid	393
8.344.2.12 MCC	393
8.344.2.13 MNC	393
8.344.2.14 networkID	393
8.344.2.15 networkIdValid	393
8.344.2.16 packetZone	393
8.344.2.17 packetZoneValid	393
8.344.2.18 pRevInUse	393
8.344.2.19 pRevInUseValid	393
8.344.2.20 sysInfoCDMA	393
8.344.2.21 systemID	393
8.345 nas_CDMASysInfoExt Struct Reference	393
8.345.1 Detailed Description	394
8.345.2 Field Documentation	394
8.345.2.1 imsi_11_12	394
8.345.2.2 MCC	394
8.346 nas_cellParams Struct Reference	394
8.346.1 Detailed Description	394
8.346.2 Field Documentation	395
8.346.2.1 pci	395
8.346.2.2 rsrp	395
8.346.2.3 rsrq	395
8.346.2.4 rssi	395
8.346.2.5 srxlev	395
8.347 nas_CommInfo Struct Reference	395
8.347.1 Detailed Description	395
8.347.2 Field Documentation	397
8.347.2.1 imsRegState	397
8.347.2.2 modemMode	397

8.347.2.3 psState . . . . .	397
8.347.2.4 systemMode . . . . .	397
8.347.2.5 temperature . . . . .	397
8.348nas_CSGID Struct Reference . . . . .	397
8.348.1 Detailed Description . . . . .	397
8.348.2 Field Documentation . . . . .	397
8.348.2.1 id . . . . .	398
8.348.2.2 mcc . . . . .	398
8.348.2.3 mnc . . . . .	398
8.348.2.4 mncPcsDigits . . . . .	398
8.348.2.5 rat . . . . .	398
8.349nas_currentPLMN Struct Reference . . . . .	398
8.349.1 Detailed Description . . . . .	398
8.349.2 Field Documentation . . . . .	398
8.349.2.1 MCC . . . . .	398
8.349.2.2 MNC . . . . .	398
8.349.2.3 netDescr . . . . .	399
8.349.2.4 netDescrLength . . . . .	399
8.350nas_dataSrvCapabilities Struct Reference . . . . .	399
8.350.1 Detailed Description . . . . .	399
8.350.2 Field Documentation . . . . .	399
8.350.2.1 dataCapabilities . . . . .	399
8.350.2.2 dataCapabilitiesLen . . . . .	399
8.351nas_detailSvcInfo Struct Reference . . . . .	399
8.351.1 Detailed Description . . . . .	400
8.351.2 Field Documentation . . . . .	401
8.351.2.1 hdrHybrid . . . . .	401
8.351.2.2 hdrSrvStatus . . . . .	401
8.351.2.3 isSysForbidden . . . . .	401
8.351.2.4 srvCapability . . . . .	401
8.351.2.5 srvStatus . . . . .	401
8.352nas_ecioListElement Struct Reference . . . . .	401
8.352.1 Detailed Description . . . . .	401
8.352.2 Field Documentation . . . . .	401
8.352.2.1 ecio . . . . .	401
8.352.2.2 radiolf . . . . .	401
8.353nas_errorRateListElement Struct Reference . . . . .	402
8.353.1 Detailed Description . . . . .	402
8.353.2 Field Documentation . . . . .	402
8.353.2.1 errorRate . . . . .	402

8.353.2.2 radiolf . . . . .	402
8.354nas_GERANInfo Struct Reference . . . . .	402
8.354.1 Detailed Description . . . . .	403
8.354.2 Field Documentation . . . . .	404
8.354.2.1 arfcn . . . . .	404
8.354.2.2 bsic . . . . .	404
8.354.2.3 cellID . . . . .	404
8.354.2.4 insNmrCellInfo . . . . .	404
8.354.2.5 lac . . . . .	404
8.354.2.6 nmrlnst . . . . .	404
8.354.2.7 plmn . . . . .	404
8.354.2.8 rxLev . . . . .	404
8.354.2.9 timingAdvance . . . . .	404
8.355nas_geranInstInfo Struct Reference . . . . .	404
8.355.1 Detailed Description . . . . .	405
8.355.2 Field Documentation . . . . .	405
8.355.2.1 geranArfcn . . . . .	405
8.355.2.2 geranBsicBcc . . . . .	405
8.355.2.3 geranBsicNcc . . . . .	405
8.355.2.4 geranRssi . . . . .	405
8.356nas_gsmCellInfo Struct Reference . . . . .	405
8.356.1 Detailed Description . . . . .	405
8.356.2 Field Documentation . . . . .	406
8.356.2.1 arfcn . . . . .	406
8.356.2.2 band1900 . . . . .	406
8.356.2.3 bsicld . . . . .	406
8.356.2.4 cellIdValid . . . . .	406
8.356.2.5 rssi . . . . .	406
8.356.2.6 srxlev . . . . .	406
8.357nas_GSMRSSIthresh Struct Reference . . . . .	406
8.357.1 Detailed Description . . . . .	406
8.357.2 Field Documentation . . . . .	407
8.357.2.1 GSMRSSIthreshListLen . . . . .	407
8.357.2.2 pGSMRSSIthreshList . . . . .	407
8.358nas_GSMSrvStatusInfo Struct Reference . . . . .	407
8.358.1 Detailed Description . . . . .	407
8.358.2 Field Documentation . . . . .	408
8.358.2.1 isPrefDataPath . . . . .	408
8.358.2.2 srvStatus . . . . .	408
8.358.2.3 trueSrvStatus . . . . .	408

8.359nas_GSMSysInfo Struct Reference . . . . .	408
8.359.1 Detailed Description . . . . .	408
8.359.2 Field Documentation . . . . .	410
8.359.2.1 cellId . . . . .	410
8.359.2.2 cellIdValid . . . . .	410
8.359.2.3 dtmSupp . . . . .	410
8.359.2.4 dtmSuppValid . . . . .	410
8.359.2.5 egprsSupp . . . . .	410
8.359.2.6 egprsSuppValid . . . . .	410
8.359.2.7 lac . . . . .	410
8.359.2.8 lacValid . . . . .	410
8.359.2.9 MCC . . . . .	410
8.359.2.10MNC . . . . .	410
8.359.2.11networkIdValid . . . . .	410
8.359.2.12regRejectInfoValid . . . . .	410
8.359.2.13rejCause . . . . .	411
8.359.2.14rejectSrvDomain . . . . .	411
8.359.2.15sysInfoGSM . . . . .	411
8.360nas_HDRECIOThresh Struct Reference . . . . .	411
8.360.1 Detailed Description . . . . .	411
8.360.2 Field Documentation . . . . .	411
8.360.2.1 HDRECIOThreshListLen . . . . .	411
8.360.2.2 pHDRECIOThreshList . . . . .	411
8.361nas_HDRIOTThresh Struct Reference . . . . .	411
8.361.1 Detailed Description . . . . .	411
8.361.2 Field Documentation . . . . .	412
8.361.2.1 HDRIOTThreshListLen . . . . .	412
8.361.2.2 pHDRIOTThreshList . . . . .	412
8.362nas_HDRRSSIThresh Struct Reference . . . . .	412
8.362.1 Detailed Description . . . . .	412
8.362.2 Field Documentation . . . . .	412
8.362.2.1 HDRRSSIThreshListLen . . . . .	412
8.362.2.2 pHDRRSSIThreshList . . . . .	412
8.363nas_HDRSINRThreshold Struct Reference . . . . .	412
8.363.1 Detailed Description . . . . .	413
8.363.2 Field Documentation . . . . .	413
8.363.2.1 HDRSINRThreshListLen . . . . .	413
8.363.2.2 pHDRSINRThreshList . . . . .	413
8.364nas_HDRSysInfo Struct Reference . . . . .	413
8.364.1 Detailed Description . . . . .	413

8.364.2 Field Documentation	415
8.364.2.1 hdrActiveProt	415
8.364.2.2 hdrActiveProtValid	415
8.364.2.3 hdrPersonality	415
8.364.2.4 hdrPersonalityValid	415
8.364.2.5 is856SysId	415
8.364.2.6 is856SysIdValid	415
8.364.2.7 isSysPrIMatch	415
8.364.2.8 isSysPrIMatchValid	415
8.364.2.9 sysInfoHDR	415
8.365nas_infoInterFreq Struct Reference	415
8.365.1 Detailed Description	415
8.365.2 Field Documentation	416
8.365.2.1 cell_resel_priority	416
8.365.2.2 cellInterFreqParams	416
8.365.2.3 cells_len	416
8.365.2.4 earfcn	416
8.365.2.5 threshXHigh	416
8.365.2.6 threshXLow	416
8.366nas_lteGsmCellInfo Struct Reference	416
8.366.1 Detailed Description	416
8.366.2 Field Documentation	417
8.366.2.1 cellReselPriority	417
8.366.2.2 cells_len	417
8.366.2.3 GsmCellInfo	417
8.366.2.4 nccPermitted	417
8.366.2.5 threshGsmHigh	417
8.366.2.6 threshGsmLow	417
8.367nas_LTEInfo Struct Reference	417
8.367.1 Detailed Description	418
8.367.2 Field Documentation	419
8.367.2.1 band	419
8.367.2.2 bandwidth	419
8.367.2.3 emmConnState	419
8.367.2.4 emmState	419
8.367.2.5 emmSubState	419
8.367.2.6 RXChan	419
8.367.2.7 TXChan	419
8.368nas_LTEInfoInterfreq Struct Reference	420
8.368.1 Detailed Description	420

8.368.2 Field Documentation	420
8.368.2.1 freqsLen	420
8.368.2.2 InfoInterfreq	420
8.368.2.3 ueInIdle	420
8.369nas_LTEInfoIntrafreq Struct Reference	420
8.369.1 Detailed Description	421
8.369.2 Field Documentation	422
8.369.2.1 CellParams	422
8.369.2.2 cellReselPriority	422
8.369.2.3 cellsLen	422
8.369.2.4 earfcn	422
8.369.2.5 globalCellId	422
8.369.2.6 plmn	422
8.369.2.7 servingCellId	422
8.369.2.8 sIntraSearch	422
8.369.2.9 sNonIntraSearch	422
8.369.2.10tac	422
8.369.2.11threshServingLow	422
8.369.2.12ueInIdle	422
8.370nas_LTEInfoNeighboringGSM Struct Reference	422
8.370.1 Detailed Description	423
8.370.2 Field Documentation	423
8.370.2.1 freqsLen	423
8.370.2.2 LteGsmCellInfo	423
8.370.2.3 ueInIdle	423
8.371nas_LTEInfoNeighboringWCDMA Struct Reference	423
8.371.1 Detailed Description	423
8.371.2 Field Documentation	424
8.371.2.1 freqsLen	424
8.371.2.2 LTEWCDMACellInfo	424
8.371.2.3 ueInIdle	424
8.372nas_LteRsrpInformation Struct Reference	424
8.372.1 Detailed Description	424
8.372.2 Field Documentation	424
8.372.2.1 rsrpLevel	424
8.373nas_LTERSRPThresh Struct Reference	424
8.373.1 Detailed Description	424
8.373.2 Field Documentation	424
8.373.2.1 LTERSRPThreshListLen	425
8.373.2.2 pLTERSRPThreshList	425

8.374nas_LTERSRQThresh Struct Reference . . . . .	425
8.374.1 Detailed Description . . . . .	425
8.374.2 Field Documentation . . . . .	425
8.374.2.1 LTERSRQThreshListLen . . . . .	425
8.374.2.2 pLTERSRQThreshList . . . . .	425
8.375nas_LTERSSIThresh Struct Reference . . . . .	425
8.375.1 Detailed Description . . . . .	425
8.375.2 Field Documentation . . . . .	426
8.375.2.1 LTERSSIThreshListLen . . . . .	426
8.375.2.2 pLTERSSIThreshList . . . . .	426
8.376nas_LTESigRptConfig Struct Reference . . . . .	426
8.376.1 Detailed Description . . . . .	426
8.376.2 Field Documentation . . . . .	426
8.376.2.1 avgPeriod . . . . .	426
8.376.2.2 rptRate . . . . .	426
8.377nas_lteSnrinformation Struct Reference . . . . .	426
8.377.1 Detailed Description . . . . .	427
8.377.2 Field Documentation . . . . .	427
8.377.2.1 snrlevel . . . . .	427
8.378nas_LTESNRThreshold Struct Reference . . . . .	427
8.378.1 Detailed Description . . . . .	427
8.378.2 Field Documentation . . . . .	427
8.378.2.1 LTESNRThreshListLen . . . . .	427
8.378.2.2 pLTESNRThreshList . . . . .	427
8.379nas_LTESysInfo Struct Reference . . . . .	427
8.379.1 Detailed Description . . . . .	428
8.379.2 Field Documentation . . . . .	429
8.379.2.1 cellId . . . . .	429
8.379.2.2 cellIdValid . . . . .	429
8.379.2.3 lac . . . . .	429
8.379.2.4 lacValid . . . . .	429
8.379.2.5 MCC . . . . .	430
8.379.2.6 MNC . . . . .	430
8.379.2.7 networkIdValid . . . . .	430
8.379.2.8 regRejectInfoValid . . . . .	430
8.379.2.9 rejCause . . . . .	430
8.379.2.10rejectSrvDomain . . . . .	430
8.379.2.11sysInfoLTE . . . . .	430
8.379.2.12ac . . . . .	430
8.379.2.13acValid . . . . .	430

8.380nas_lteWcdmaCellInfo Struct Reference	430
8.380.1 Detailed Description	430
8.380.2 Field Documentation	431
8.380.2.1 cellReselPriority	431
8.380.2.2 cellsLen	431
8.380.2.3 threshXhigh	431
8.380.2.4 threshXlow	431
8.380.2.5 uarfcn	431
8.380.2.6 WCDMACellInfo	431
8.381nas_MNRInfo Struct Reference	431
8.381.1 Detailed Description	431
8.381.2 Field Documentation	431
8.381.2.1 mcc	432
8.381.2.2 mnc	432
8.381.2.3 rat	432
8.382nas_netSelectionPref Struct Reference	432
8.382.1 Detailed Description	432
8.382.2 Field Documentation	432
8.382.2.1 mcc	432
8.382.2.2 mnc	432
8.382.2.3 netReg	432
8.383nas_nmrCellInfo Struct Reference	432
8.383.1 Detailed Description	433
8.383.2 Field Documentation	433
8.383.2.1 nmrArfcn	433
8.383.2.2 nmrBsic	433
8.383.2.3 nmrCellID	433
8.383.2.4 nmrLac	433
8.383.2.5 nmrPlmn	433
8.383.2.6 nmrRxLev	433
8.384nas_PhyCaAggPcellInfo Struct Reference	434
8.384.1 Detailed Description	434
8.384.2 Field Documentation	434
8.384.2.1 dl_bw_value	434
8.384.2.2 freq	434
8.384.2.3 iLTEbandValue	434
8.384.2.4 pci	434
8.384.2.5 TlvPresent	434
8.385nas_PhyCaAggScellDIBw Struct Reference	434
8.385.1 Detailed Description	435

8.385.2 Field Documentation . . . . .	435
8.385.2.1 dl_bw_value . . . . .	435
8.385.2.2 TlvPresent . . . . .	435
8.386nas_PhyCaAggScellIndex Struct Reference . . . . .	435
8.386.1 Detailed Description . . . . .	435
8.386.2 Field Documentation . . . . .	435
8.386.2.1 scell_idx . . . . .	435
8.386.2.2 TlvPresent . . . . .	435
8.387nas_PhyCaAggScellIndType Struct Reference . . . . .	435
8.387.1 Detailed Description . . . . .	436
8.387.2 Field Documentation . . . . .	436
8.387.2.1 freq . . . . .	436
8.387.2.2 pci . . . . .	436
8.387.2.3 scell_state . . . . .	436
8.387.2.4 TlvPresent . . . . .	436
8.388nas_PhyCaAggScellInfo Struct Reference . . . . .	436
8.388.1 Detailed Description . . . . .	436
8.388.2 Field Documentation . . . . .	439
8.388.2.1 dl_bw_value . . . . .	439
8.388.2.2 freq . . . . .	439
8.388.2.3 iLTEbandValue . . . . .	439
8.388.2.4 pci . . . . .	439
8.388.2.5 scell_state . . . . .	439
8.388.2.6 TlvPresent . . . . .	439
8.389nas_qaQmi3Gpp2TimeZone Struct Reference . . . . .	439
8.389.1 Detailed Description . . . . .	439
8.389.2 Field Documentation . . . . .	439
8.389.2.1 daylightSavings . . . . .	439
8.389.2.2 leapSeconds . . . . .	439
8.389.2.3 localTimeOffset . . . . .	440
8.390nas_QmiNas3GppNetworkInfo Struct Reference . . . . .	440
8.390.1 Detailed Description . . . . .	440
8.390.2 Field Documentation . . . . .	440
8.390.2.1 Desription . . . . .	440
8.390.2.2 Forbidden . . . . .	440
8.390.2.3 InUse . . . . .	440
8.390.2.4 MCC . . . . .	440
8.390.2.5 MNC . . . . .	440
8.390.2.6 Preferred . . . . .	440
8.390.2.7 Roaming . . . . .	440

8.391nas_QmiNas3GppNetworkRAT Struct Reference	440
8.391.1 Detailed Description	440
8.391.2 Field Documentation	441
8.391.2.1 MCC	441
8.391.2.2 MNC	441
8.391.2.3 RAT	441
8.392nas_QmisNasPcsDigit Struct Reference	441
8.392.1 Detailed Description	441
8.392.2 Field Documentation	441
8.392.2.1 includes_pcs_digit	442
8.392.2.2 MCC	442
8.392.2.3 MNC	442
8.393nas_RejectReasonTlv Struct Reference	442
8.393.1 Detailed Description	442
8.393.2 Field Documentation	442
8.393.2.1 rejectCause	442
8.393.2.2 serviceDomain	442
8.393.2.3 TlvPresent	442
8.394nas_RFInfoTlv Struct Reference	442
8.394.1 Detailed Description	442
8.394.2 Field Documentation	442
8.394.2.1 activeBandClass	442
8.394.2.2 activeChannel	443
8.394.2.3 radiolInterface	443
8.394.2.4 radiolInterfaceSize	443
8.394.2.5 TlvPresent	443
8.395nas_roamIndList Struct Reference	443
8.395.1 Detailed Description	443
8.395.2 Field Documentation	443
8.395.2.1 numInstances	443
8.395.2.2 radiolInterface	443
8.395.2.3 roamIndicator	444
8.396nas_rsrqInformation Struct Reference	444
8.396.1 Detailed Description	444
8.396.2 Field Documentation	444
8.396.2.1 radiolf	444
8.396.2.2 rsrq	444
8.397nas_rxSignalStrengthListElement Struct Reference	444
8.397.1 Detailed Description	444
8.397.2 Field Documentation	445

8.397.2.1 radiolf . . . . .	445
8.397.2.2 rxSignalStrength . . . . .	445
8.398nas_servSystem Struct Reference . . . . .	445
8.398.1 Detailed Description . . . . .	445
8.398.2 Field Documentation . . . . .	446
8.398.2.1 csAttachState . . . . .	446
8.398.2.2 numRadioInterfaces . . . . .	446
8.398.2.3 psAttachState . . . . .	446
8.398.2.4 radiolInterface . . . . .	446
8.398.2.5 regState . . . . .	446
8.398.2.6 selNetwork . . . . .	446
8.399nas_SignalStrengthTlv Struct Reference . . . . .	447
8.399.1 Detailed Description . . . . .	447
8.399.2 Field Documentation . . . . .	447
8.399.2.1 radiolInterface . . . . .	447
8.399.2.2 signalStrength . . . . .	447
8.399.2.3 TlvPresent . . . . .	447
8.400nas_SLQSSignalStrengthsIndReq Struct Reference . . . . .	447
8.400.1 Detailed Description . . . . .	447
8.400.2 Field Documentation . . . . .	448
8.400.2.1 ecioDelta . . . . .	448
8.400.2.2 ecioThresholdList . . . . .	448
8.400.2.3 ecioThresholdListLen . . . . .	448
8.400.2.4 ioDelta . . . . .	448
8.400.2.5 lteRsrpDelta . . . . .	448
8.400.2.6 lteSnrDelta . . . . .	448
8.400.2.7 rsrqDelta . . . . .	448
8.400.2.8 rxSignalStrengthDelta . . . . .	448
8.400.2.9 sinrDelta . . . . .	448
8.400.2.10sinrThresholdList . . . . .	448
8.400.2.11sinrThresholdListLen . . . . .	448
8.401nas_SLQSSignalStrengthsInformation Struct Reference . . . . .	448
8.401.1 Detailed Description . . . . .	448
8.401.2 Field Documentation . . . . .	449
8.401.2.1 ecioInfo . . . . .	449
8.401.2.2 errorRateInfo . . . . .	449
8.401.2.3 io . . . . .	449
8.401.2.4 lteRsrpinfo . . . . .	449
8.401.2.5 lteSnrinfo . . . . .	449
8.401.2.6 rsrqInfo . . . . .	449

8.401.2.7 rxSignalStrengthInfo . . . . .	449
8.401.2.8 sinr . . . . .	449
8.402nas_SLQSSignalStrengthsTlv Struct Reference . . . . .	449
8.402.1 Detailed Description . . . . .	449
8.402.2 Field Documentation . . . . .	449
8.402.2.1 sSLQSSignalStrengthsInfo . . . . .	449
8.402.2.2 TlvPresent . . . . .	449
8.403nas_SrvStatusInfo Struct Reference . . . . .	449
8.403.1 Detailed Description . . . . .	450
8.403.2 Field Documentation . . . . .	450
8.403.2.1 isPrefDataPath . . . . .	450
8.403.2.2 srvStatus . . . . .	450
8.404nas_sysInfoCommon Struct Reference . . . . .	450
8.404.1 Detailed Description . . . . .	450
8.404.2 Field Documentation . . . . .	452
8.404.2.1 isSysForbidden . . . . .	452
8.404.2.2 isSysForbiddenValid . . . . .	452
8.404.2.3 roamStatus . . . . .	452
8.404.2.4 roamStatusValid . . . . .	452
8.404.2.5 srvCapability . . . . .	452
8.404.2.6 srvCapabilityValid . . . . .	452
8.404.2.7 srvDomain . . . . .	452
8.404.2.8 srvDomainValid . . . . .	452
8.405nas_TDSCDMAECIOThresh Struct Reference . . . . .	453
8.405.1 Detailed Description . . . . .	453
8.405.2 Field Documentation . . . . .	453
8.405.2.1 pTDSCDMAECIOThreshList . . . . .	453
8.405.2.2 TDSCDMAECIOThreshListLen . . . . .	453
8.406nas_TDSCDMARSCPThresh Struct Reference . . . . .	453
8.406.1 Detailed Description . . . . .	453
8.406.2 Field Documentation . . . . .	453
8.406.2.1 pTDSCDMARSCPThreshList . . . . .	453
8.406.2.2 TDSCDMARSCPThreshListLen . . . . .	454
8.407nas_TDSCDMARSSIThresh Struct Reference . . . . .	454
8.407.1 Detailed Description . . . . .	454
8.407.2 Field Documentation . . . . .	454
8.407.2.1 pTDSCDMARSSIThreshList . . . . .	454
8.407.2.2 TDSCDMARSSIThreshListLen . . . . .	454
8.408nas_TDSCDMASINRThresh Struct Reference . . . . .	454
8.408.1 Detailed Description . . . . .	454

8.408.2 Field Documentation	454
8.408.2.1 pTDSCDMASINRThreshList	455
8.408.2.2 TDSCDMASINRThreshListLen	455
8.409nas_timeInfo Struct Reference	455
8.409.1 Detailed Description	455
8.409.2 Field Documentation	456
8.409.2.1 day	456
8.409.2.2 dayLtSavingAdj	456
8.409.2.3 dayOfWeek	456
8.409.2.4 hour	456
8.409.2.5 minute	456
8.409.2.6 month	456
8.409.2.7 radiolInterface	456
8.409.2.8 second	456
8.409.2.9 timeZone	456
8.409.2.10TlvPresent	456
8.409.2.11year	456
8.410nas_UMTSInfo Struct Reference	456
8.410.1 Detailed Description	457
8.410.2 Field Documentation	458
8.410.2.1 cellID	458
8.410.2.2 ecio	458
8.410.2.3 geranInst	458
8.410.2.4 GeranInstInfo	458
8.410.2.5 lac	458
8.410.2.6 plmn	458
8.410.2.7 psc	458
8.410.2.8 rscp	458
8.410.2.9 uarfcn	458
8.410.2.10umtsInst	458
8.410.2.11UMTSInstInfo	458
8.411nas_UMTSinstInfo Struct Reference	458
8.411.1 Detailed Description	458
8.411.2 Field Documentation	459
8.411.2.1 umtsEcio	459
8.411.2.2 umtsPsc	459
8.411.2.3 umtsRscp	459
8.411.2.4 umtsUarfcn	459
8.412nas_umtsLTENbrCell Struct Reference	459
8.412.1 Detailed Description	459

8.412.2 Field Documentation . . . . .	460
8.412.2.1 cellIsTDD . . . . .	460
8.412.2.2 earfcn . . . . .	460
8.412.2.3 pci . . . . .	460
8.412.2.4 rsrp . . . . .	460
8.412.2.5 rsrq . . . . .	460
8.412.2.6 srxlev . . . . .	460
8.413nas_UniversalTime Struct Reference . . . . .	460
8.413.1 Detailed Description . . . . .	460
8.413.2 Field Documentation . . . . .	461
8.413.2.1 day . . . . .	461
8.413.2.2 dayOfWeek . . . . .	461
8.413.2.3 hour . . . . .	461
8.413.2.4 minute . . . . .	461
8.413.2.5 month . . . . .	461
8.413.2.6 second . . . . .	461
8.413.2.7 year . . . . .	461
8.414nas_wcdmaCellInfo Struct Reference . . . . .	461
8.414.1 Detailed Description . . . . .	461
8.414.2 Field Documentation . . . . .	462
8.414.2.1 cpich_ecno . . . . .	462
8.414.2.2 cpich_rscp . . . . .	462
8.414.2.3 psc . . . . .	462
8.414.2.4 srxlev . . . . .	462
8.415nas_WCDMAECIOThresh Struct Reference . . . . .	462
8.415.1 Detailed Description . . . . .	462
8.415.2 Field Documentation . . . . .	462
8.415.2.1 pWCDMAECIOThreshList . . . . .	462
8.415.2.2 WCDMAECIOThreshListLen . . . . .	463
8.416nas_WCDMAInfoLTENeighborCell Struct Reference . . . . .	463
8.416.1 Detailed Description . . . . .	463
8.416.2 Field Documentation . . . . .	463
8.416.2.1 UMTSLTENbrCell . . . . .	463
8.416.2.2 umtsLTENbrCellLen . . . . .	463
8.416.2.3 wcdmaRRCState . . . . .	463
8.417nas_WCDMARSSIThresh Struct Reference . . . . .	463
8.417.1 Detailed Description . . . . .	464
8.417.2 Field Documentation . . . . .	464
8.417.2.1 pWCDMARSSIThreshList . . . . .	464
8.417.2.2 WCDMARSSIThreshListLen . . . . .	464

8.418nas_WCDMASysInfo Struct Reference . . . . .	464
8.418.1 Detailed Description . . . . .	464
8.418.2 Field Documentation . . . . .	467
8.418.2.1 cellId . . . . .	467
8.418.2.2 cellIdValid . . . . .	467
8.418.2.3 hsCallStatus . . . . .	467
8.418.2.4 hsCallStatusValid . . . . .	467
8.418.2.5 hsInd . . . . .	467
8.418.2.6 hsIndValid . . . . .	467
8.418.2.7 lac . . . . .	467
8.418.2.8 lacValid . . . . .	467
8.418.2.9 MCC . . . . .	467
8.418.2.10MNC . . . . .	467
8.418.2.11networkIdValid . . . . .	467
8.418.2.12psc . . . . .	467
8.418.2.13pscValid . . . . .	467
8.418.2.14regRejectInfoValid . . . . .	467
8.418.2.15rejCause . . . . .	467
8.418.2.16rejectSrvDomain . . . . .	467
8.418.2.17sysInfoWCDMA . . . . .	467
8.419NASBandPreferenceTlv Struct Reference . . . . .	467
8.419.1 Field Documentation . . . . .	467
8.419.1.1 band_pref . . . . .	467
8.419.1.2 TlvPresent . . . . .	468
8.420nasCellLocationInfoResp Struct Reference . . . . .	468
8.420.1 Detailed Description . . . . .	468
8.420.2 Field Documentation . . . . .	468
8.420.2.1 pCDMAInfo . . . . .	469
8.420.2.2 pGERANInfo . . . . .	469
8.420.2.3 pLTEInfoInterfreq . . . . .	469
8.420.2.4 pLTEInfoIntrafreq . . . . .	469
8.420.2.5 pLTEInfoNeighboringGSM . . . . .	469
8.420.2.6 pLTEInfoNeighboringWCDMA . . . . .	469
8.420.2.7 pUMTSCellID . . . . .	469
8.420.2.8 pUMTSInfo . . . . .	469
8.420.2.9 pWCDMAInfoLTENeighborCell . . . . .	469
8.421NASEmergencyModeTlv Struct Reference . . . . .	469
8.421.1 Field Documentation . . . . .	469
8.421.1.1 EmerMode . . . . .	469
8.421.1.2 TlvPresent . . . . .	469

8.422nasGet3GPP2SubscriptionInfoReq Struct Reference . . . . .	469
8.422.1 Detailed Description . . . . .	469
8.422.2 Field Documentation . . . . .	469
8.422.2.1 namID . . . . .	469
8.423nasGet3GPP2SubscriptionInfoResp Struct Reference . . . . .	470
8.423.1 Detailed Description . . . . .	470
8.423.2 Field Documentation . . . . .	470
8.423.2.1 pCDMAChannel . . . . .	470
8.423.2.2 pDirNum . . . . .	470
8.423.2.3 pHomeSIDNID . . . . .	470
8.423.2.4 pMinBasedIMSI . . . . .	470
8.423.2.5 pNAMNameInfo . . . . .	470
8.423.2.6 pTrueIMSI . . . . .	470
8.424nasGetHDRColorCodeResp Struct Reference . . . . .	470
8.424.1 Detailed Description . . . . .	471
8.424.2 Field Documentation . . . . .	471
8.424.2.1 pColorCode . . . . .	471
8.425nasGetLTECphyCa Struct Reference . . . . .	471
8.425.1 Field Documentation . . . . .	471
8.425.1.1 sPhyCaAggPcellInfo . . . . .	471
8.425.1.2 sPhyCaAggScellIDBw . . . . .	471
8.425.1.3 sPhyCaAggScellIndex . . . . .	471
8.425.1.4 sPhyCaAggScellIndType . . . . .	471
8.425.1.5 sPhyCaAggScellInfo . . . . .	471
8.426NasGetLTECphyCaInfo Struct Reference . . . . .	471
8.426.1 Field Documentation . . . . .	472
8.426.1.1 PhyCaAggPcellInfo . . . . .	472
8.426.1.2 PhyCaAggScellIDBw . . . . .	472
8.426.1.3 PhyCaAggScellIndex . . . . .	472
8.426.1.4 PhyCaAggScellIndType . . . . .	472
8.426.1.5 PhyCaAggScellInfo . . . . .	472
8.427nasGetLTECphyCaResp Struct Reference . . . . .	472
8.427.1 Field Documentation . . . . .	472
8.427.1.1 pPhyCaAggPcellInfo . . . . .	472
8.427.1.2 pPhyCaAggScellIDBw . . . . .	472
8.427.1.3 pPhyCaAggScellIndex . . . . .	472
8.427.1.4 pPhyCaAggScellIndType . . . . .	472
8.427.1.5 pPhyCaAggScellInfo . . . . .	472
8.428nasGetSigInfoResp Struct Reference . . . . .	472
8.428.1 Detailed Description . . . . .	472

8.428.2 Field Documentation	473
8.428.2.1 pCDMASSInfo	473
8.428.2.2 pGSMSSInfo	473
8.428.2.3 pHDRSSInfo	473
8.428.2.4 pLTESSInfo	473
8.428.2.5 pTDSCDMASigInfoExt	473
8.428.2.6 pTDSCDMASigInfoRscp	473
8.428.2.7 pWCDMASSInfo	473
8.429nasGetSysInfoResp Struct Reference	473
8.429.1 Detailed Description	474
8.429.2 Field Documentation	475
8.429.2.1 pAddCDMASysInfo	475
8.429.2.2 pAddGSMSysInfo	475
8.429.2.3 pAddHDRSysInfo	475
8.429.2.4 pAddLTESysInfo	475
8.429.2.5 pAddWCDMASysInfo	475
8.429.2.6 pCDMASrvStatusInfo	475
8.429.2.7 pCDMASysInfo	475
8.429.2.8 pGSMCallBarringSysInfo	476
8.429.2.9 pGSMCipherDomainSysInfo	476
8.429.2.10pGSMSrvStatusInfo	476
8.429.2.11pGSMSysInfo	476
8.429.2.12pHDRSrvStatusInfo	476
8.429.2.13pHDRSysInfo	476
8.429.2.14pLTESrvStatusInfo	476
8.429.2.15pLTESysInfo	476
8.429.2.16pLTEVoiceSupportSysInfo	476
8.429.2.17pWCDMACallBarringSysInfo	476
8.429.2.18pWCDMACipherDomainSysInfo	476
8.429.2.19pWCDMASrvStatusInfo	476
8.429.2.20pWCDMASysInfo	476
8.430nasGetTxRxInfoReq Struct Reference	476
8.430.1 Detailed Description	476
8.430.2 Field Documentation	476
8.430.2.1 radio_if	476
8.431nasGetTxRxInfoResp Struct Reference	477
8.431.1 Detailed Description	477
8.431.2 Field Documentation	477
8.431.2.1 pRXChain0Info	477
8.431.2.2 pRXChain1Info	477

8.431.2.3 pTXInfo . . . . .	477
8.432NASGWAcqOrderPrefTlv Struct Reference . . . . .	477
8.432.1 Field Documentation . . . . .	477
8.432.1.1 GWAcqOrderPref . . . . .	477
8.432.1.2 TlvPresent . . . . .	477
8.433nasIndicationRegisterReq Struct Reference . . . . .	477
8.433.1 Detailed Description . . . . .	478
8.433.2 Field Documentation . . . . .	480
8.433.2.1 pDDTMInd . . . . .	480
8.433.2.2 pDualStandByPrefInd . . . . .	480
8.433.2.3 pErrorRateInd . . . . .	480
8.433.2.4 pHDRNewUATIAssInd . . . . .	480
8.433.2.5 pHDRSessionCloseInd . . . . .	480
8.433.2.6 pLTECphyCa . . . . .	480
8.433.2.7 pManagedRoamingInd . . . . .	480
8.433.2.8 pNetworkTimeInd . . . . .	480
8.433.2.9 pServingSystemInd . . . . .	480
8.433.2.10pSignalStrengthInd . . . . .	480
8.433.2.11pSubscriptionInfoInd . . . . .	480
8.433.2.12pSysInfoInd . . . . .	480
8.433.2.13pSystemSelectionInd . . . . .	480
8.434nasInitNetworkReg Struct Reference . . . . .	480
8.434.1 Detailed Description . . . . .	480
8.434.2 Field Documentation . . . . .	481
8.434.2.1 pChangeDuration . . . . .	481
8.434.2.2 pMncPcsDigitStatus . . . . .	481
8.434.2.3 pMNRInfo . . . . .	481
8.434.2.4 regAction . . . . .	481
8.435NASLTEBandPreferenceTlv Struct Reference . . . . .	481
8.435.1 Field Documentation . . . . .	481
8.435.1.1 LTEBandPref . . . . .	481
8.435.1.2 TlvPresent . . . . .	481
8.436NASLteNasReleaseInfoTlv Struct Reference . . . . .	481
8.436.1 Field Documentation . . . . .	481
8.436.1.1 nas_major . . . . .	481
8.436.1.2 nas_minor . . . . .	481
8.436.1.3 nas_release . . . . .	481
8.436.1.4 TlvPresent . . . . .	482
8.437NASModePreferenceTlv Struct Reference . . . . .	482
8.437.1 Field Documentation . . . . .	482

8.437.1.1 ModePref . . . . .	482
8.437.1.2 TlvPresent . . . . .	482
8.438NASNetSelPreferenceTlv Struct Reference . . . . .	482
8.438.1 Field Documentation . . . . .	482
8.438.1.1 NetSelPref . . . . .	482
8.438.1.2 TlvPresent . . . . .	482
8.439nasNetworkTime Struct Reference . . . . .	482
8.439.1 Detailed Description . . . . .	482
8.439.2 Field Documentation . . . . .	483
8.439.2.1 pDayltSavAdj . . . . .	483
8.439.2.2 pRadioInterface . . . . .	483
8.439.2.3 pTimeZone . . . . .	483
8.439.2.4 universalTime . . . . .	483
8.440nasOperatorNameResp Struct Reference . . . . .	483
8.440.1 Detailed Description . . . . .	483
8.440.2 Field Documentation . . . . .	484
8.440.2.1 pNITZInformation . . . . .	484
8.440.2.2 pOperatorNameString . . . . .	484
8.440.2.3 pOperatorPLMNList . . . . .	484
8.440.2.4 pPLMNNetworkName . . . . .	484
8.440.2.5 pSrvProviderName . . . . .	484
8.441NASOTAMessageTlv Struct Reference . . . . .	484
8.441.1 Field Documentation . . . . .	484
8.441.1.1 data_buf . . . . .	484
8.441.1.2 data_len . . . . .	484
8.441.1.3 message_type . . . . .	484
8.441.1.4 TlvPresent . . . . .	484
8.442NASPhyCaAggPcellInfo Struct Reference . . . . .	484
8.442.1 Detailed Description . . . . .	485
8.442.2 Field Documentation . . . . .	485
8.442.2.1 dl_bw_value . . . . .	485
8.442.2.2 freq . . . . .	485
8.442.2.3 iLTEbandValue . . . . .	485
8.442.2.4 pci . . . . .	485
8.442.2.5 TlvPresent . . . . .	485
8.443NASPhyCaAggScellIDIBw Struct Reference . . . . .	485
8.443.1 Detailed Description . . . . .	485
8.443.2 Field Documentation . . . . .	486
8.443.2.1 dl_bw_value . . . . .	486
8.443.2.2 TlvPresent . . . . .	486

8.444NASPhyCaAggScellIndex Struct Reference	486
8.444.1 Detailed Description	486
8.444.2 Field Documentation	486
8.444.2.1 scell_idx	486
8.444.2.2 TlvPresent	486
8.445NASPhyCaAggScellIndType Struct Reference	486
8.445.1 Detailed Description	486
8.445.2 Field Documentation	487
8.445.2.1 freq	487
8.445.2.2 pci	487
8.445.2.3 scell_state	487
8.445.2.4 TlvPresent	487
8.446NASPhyCaAggScellInfo Struct Reference	487
8.446.1 Detailed Description	487
8.446.2 Field Documentation	488
8.446.2.1 dl_bw_value	488
8.446.2.2 freq	488
8.446.2.3 iLTEbandValue	488
8.446.2.4 pci	488
8.446.2.5 scell_state	488
8.446.2.6 TlvPresent	488
8.447nasPLMNNameReq Struct Reference	488
8.447.1 Detailed Description	488
8.447.2 Field Documentation	489
8.447.2.1 mcc	489
8.447.2.2 mnc	489
8.447.2.3 pMncPcsStatus	489
8.448nasPLMNNameResp Struct Reference	489
8.448.1 Detailed Description	489
8.448.2 Field Documentation	491
8.448.2.1 longName	491
8.448.2.2 longNameCI	491
8.448.2.3 longNameEn	491
8.448.2.4 longNameLen	491
8.448.2.5 longNameSB	491
8.448.2.6 shortName	491
8.448.2.7 shortNameCI	491
8.448.2.8 shortNameEn	491
8.448.2.9 shortNameLen	491
8.448.2.10shortNameSB	491

8.448.2.11spn . . . . .	491
8.448.2.12spnEncoding . . . . .	491
8.448.2.13spnLength . . . . .	491
8.449NASPRLPreferenceTlv Struct Reference . . . . .	491
8.449.1 Field Documentation . . . . .	492
8.449.1.1 PRLPref . . . . .	492
8.449.1.2 TlvPresent . . . . .	492
8.450NASQmiCbkNasSwiOTAMessageInd Struct Reference . . . . .	492
8.450.1 Field Documentation . . . . .	492
8.450.1.1 nasRelInfoTlv . . . . .	492
8.450.1.2 otaMsgTlv . . . . .	492
8.450.1.3 timeTlv . . . . .	492
8.451NASQmiCbkNasSystemSelPrefInd Struct Reference . . . . .	492
8.451.1 Field Documentation . . . . .	492
8.451.1.1 BPTlv . . . . .	492
8.451.1.2 EMTlv . . . . .	492
8.451.1.3 GWAOPTlv . . . . .	492
8.451.1.4 LBPTlv . . . . .	492
8.451.1.5 MPTlv . . . . .	492
8.451.1.6 NSPTlv . . . . .	493
8.451.1.7 PRLPTlv . . . . .	493
8.451.1.8 RPTlv . . . . .	493
8.451.1.9 SDPTlv . . . . .	493
8.452NASRoamPreferenceTlv Struct Reference . . . . .	493
8.452.1 Field Documentation . . . . .	493
8.452.1.1 RoamPref . . . . .	493
8.452.1.2 TlvPresent . . . . .	493
8.453NASServDomainPrefTlv Struct Reference . . . . .	493
8.453.1 Field Documentation . . . . .	493
8.453.1.1 SrvDomainPref . . . . .	493
8.453.1.2 TlvPresent . . . . .	493
8.454NASServingSystemInfo Struct Reference . . . . .	493
8.454.1 Detailed Description . . . . .	493
8.454.2 Field Documentation . . . . .	494
8.454.2.1 csAttachState . . . . .	494
8.454.2.2 hdrPersonality . . . . .	494
8.454.2.3 psAttachState . . . . .	494
8.454.2.4 radiolInterfaceList . . . . .	495
8.454.2.5 radiolInterfaceNo . . . . .	495
8.454.2.6 registrationState . . . . .	495

8.454.2.7 selectedNetwork . . . . .	495
8.455nasSigInfo Struct Reference . . . . .	495
8.455.1 Detailed Description . . . . .	495
8.455.2 Field Documentation . . . . .	495
8.455.2.1 pCDMASigInfo . . . . .	495
8.455.2.2 pGSMSigInfo . . . . .	496
8.455.2.3 pHDRSigInfo . . . . .	496
8.455.2.4 pLTSigInfo . . . . .	496
8.455.2.5 pRscp . . . . .	496
8.455.2.6 pTDSCDMASigInfoExt . . . . .	496
8.455.2.7 pWCDMASigInfo . . . . .	496
8.456nasSwiGetChannelLockResp Struct Reference . . . . .	496
8.456.1 Detailed Description . . . . .	496
8.456.2 Field Documentation . . . . .	496
8.456.2.1 pLteEARFCN . . . . .	496
8.456.2.2 pLtePCI . . . . .	496
8.456.2.3 pWcdmaUARFCN . . . . .	496
8.457NasSwiIndReg Struct Reference . . . . .	496
8.457.1 Detailed Description . . . . .	497
8.457.2 Field Documentation . . . . .	497
8.457.2.1 gsmUmtsDI . . . . .	497
8.457.2.2 gsmUmtsUI . . . . .	497
8.457.2.3 lteEmmDI . . . . .	497
8.457.2.4 lteEmmUI . . . . .	497
8.457.2.5 lteEsmDI . . . . .	497
8.457.2.6 lteEsmUI . . . . .	497
8.457.2.7 pRankIndicatorInd . . . . .	497
8.458nasSwiSetChannelLockReq Struct Reference . . . . .	498
8.458.1 Detailed Description . . . . .	498
8.458.2 Field Documentation . . . . .	498
8.458.2.1 pLteEARFCN . . . . .	498
8.458.2.2 pLtePCI . . . . .	498
8.458.2.3 pWcdmaUARFCN . . . . .	498
8.459nasSysInfo Struct Reference . . . . .	498
8.459.1 Detailed Description . . . . .	499
8.459.2 Field Documentation . . . . .	500
8.459.2.1 pAddCDMASysInfo . . . . .	500
8.459.2.2 pAddGSMSysInfo . . . . .	500
8.459.2.3 pAddHDRSysInfo . . . . .	500
8.459.2.4 pAddLTSysInfo . . . . .	500

8.459.2.5 pAddWCDMASysInfo . . . . .	500
8.459.2.6 pCDMASrvStatusInfo . . . . .	500
8.459.2.7 pCDMASysInfo . . . . .	500
8.459.2.8 pGSMCallBarringSysInfo . . . . .	501
8.459.2.9 pGSMCipherDomainSysInfo . . . . .	501
8.459.2.10pGSMSrvStatusInfo . . . . .	501
8.459.2.11pGSMSysInfo . . . . .	501
8.459.2.12pHDRSrvStatusInfo . . . . .	501
8.459.2.13pHDRSysInfo . . . . .	501
8.459.2.14pLTESrvStatusInfo . . . . .	501
8.459.2.15pLTESysInfo . . . . .	501
8.459.2.16pLTEVoiceSupportSysInfo . . . . .	501
8.459.2.17pSysInfoNoChange . . . . .	501
8.459.2.18pWCDMACallBarringSysInfo . . . . .	501
8.459.2.19pWCDMACipherDomainSysInfo . . . . .	501
8.459.2.20pWCDMASrvStatusInfo . . . . .	501
8.459.2.21pWCDMASysInfo . . . . .	501
8.460NASTimeInfoTlv Struct Reference . . . . .	501
8.460.1 Field Documentation . . . . .	501
8.460.1.1 time . . . . .	501
8.460.1.2 TlvPresent . . . . .	501
8.461netSelectionPref Struct Reference . . . . .	501
8.461.1 Detailed Description . . . . .	501
8.461.2 Field Documentation . . . . .	502
8.461.2.1 mcc . . . . .	502
8.461.2.2 mnc . . . . .	502
8.461.2.3 netReg . . . . .	502
8.462NetStats Struct Reference . . . . .	502
8.462.1 Detailed Description . . . . .	502
8.462.2 Field Documentation . . . . .	503
8.462.2.1 rx_bytes . . . . .	503
8.462.2.2 rx_errors . . . . .	503
8.462.2.3 rx_overflows . . . . .	503
8.462.2.4 rx_packets . . . . .	503
8.462.2.5 tx_bytes . . . . .	503
8.462.2.6 tx_errors . . . . .	503
8.462.2.7 tx_overflows . . . . .	503
8.462.2.8 tx_packets . . . . .	503
8.463NetworkDebugResp Struct Reference . . . . .	503
8.463.1 Detailed Description . . . . .	503

8.463.2 Field Documentation	504
8.463.2.1 pDataStatusDetail	504
8.463.2.2 pDeviceConfigDetail	504
8.463.2.3 pNetworkStat1x	504
8.463.2.4 pNetworkStatEVDO	504
8.463.2.5 pObjectVer	504
8.464NetworkStat1x Struct Reference	504
8.464.1 Detailed Description	504
8.464.2 Field Documentation	506
8.464.2.1 ActSetCnt	506
8.464.2.2 NeighborSetCnt	506
8.464.2.3 pActPilotPNElements	506
8.464.2.4 pNeighborSetPilotPN	506
8.464.2.5 RX_EC_IO	506
8.464.2.6 RX_PWR	506
8.464.2.7 SO	506
8.464.2.8 State	506
8.464.2.9 TX_PWR	506
8.465NetworkStatEVDO Struct Reference	506
8.465.1 Detailed Description	506
8.465.2 Field Documentation	507
8.465.2.1 MACIndex	508
8.465.2.2 PER	508
8.465.2.3 PilotEnergy	508
8.465.2.4 pSectorID	508
8.465.2.5 RX_PWR	508
8.465.2.6 SectorIDLen	508
8.465.2.7 SNR	508
8.465.2.8 State	508
8.466newMTMessageTlv Struct Reference	508
8.466.1 Detailed Description	508
8.466.2 Field Documentation	508
8.466.2.1 MTMessageInfo	508
8.466.2.2 TlvPresent	508
8.467newPwdData Struct Reference	508
8.467.1 Detailed Description	508
8.467.2 Field Documentation	509
8.467.2.1 newPwd	509
8.467.2.2 newPwdAgain	509
8.468nmrCellInfo Struct Reference	509

8.468.1 Detailed Description	509
8.468.2 Field Documentation	510
8.468.2.1 nmrArfcn	510
8.468.2.2 nmrBsic	510
8.468.2.3 nmrCellID	510
8.468.2.4 nmrLac	510
8.468.2.5 nmrPlmn	510
8.468.2.6 nmrRxLev	510
8.469NSSAudioCtrl Struct Reference	510
8.469.1 Detailed Description	510
8.469.2 Field Documentation	511
8.469.2.1 downLink	511
8.469.2.2 upLink	511
8.470NWProfile Struct Reference	511
8.470.1 Detailed Description	511
8.470.2 Field Documentation	511
8.470.2.1 pProfSz	511
8.470.2.2 pProfValues	511
8.470.2.3 tech	511
8.471omaDmConfigTlv Struct Reference	511
8.471.1 Detailed Description	511
8.471.2 Field Documentation	512
8.471.2.1 alertmsg	512
8.471.2.2 alertmsglength	512
8.471.2.3 state	512
8.471.2.4 userInputReq	512
8.471.2.5 userInputTimeout	512
8.472omaDmConfigTlvExt Struct Reference	512
8.472.1 Detailed Description	512
8.472.2 Field Documentation	514
8.472.2.1 alertmsg	514
8.472.2.2 alertmsglength	514
8.472.2.3 state	514
8.472.2.4 userInputReq	514
8.472.2.5 userInputTimeout	514
8.473omaDmFotaTlv Struct Reference	514
8.473.1 Detailed Description	514
8.473.2 Field Documentation	515
8.473.2.1 description	515
8.473.2.2 descriptionlength	515

8.473.2.3 fwdloadsize . . . . .	515
8.473.2.4 fwloadComplete . . . . .	516
8.473.2.5 namelength . . . . .	516
8.473.2.6 package_name . . . . .	516
8.473.2.7 sessionType . . . . .	516
8.473.2.8 severity . . . . .	516
8.473.2.9 state . . . . .	516
8.473.2.10updateCompleteStatus . . . . .	516
8.473.2.11userInputReq . . . . .	516
8.473.2.12userInputTimeout . . . . .	516
8.473.2.13version . . . . .	516
8.473.2.14versionlength . . . . .	516
8.474omaDmFotaTlvExt Struct Reference . . . . .	516
8.474.1 Detailed Description . . . . .	516
8.474.2 Field Documentation . . . . .	518
8.474.2.1 description . . . . .	518
8.474.2.2 descriptionlength . . . . .	518
8.474.2.3 fumoResultCode . . . . .	518
8.474.2.4 namelength . . . . .	518
8.474.2.5 package_name . . . . .	518
8.474.2.6 packageSize . . . . .	518
8.474.2.7 receivedBytes . . . . .	518
8.474.2.8 reserved . . . . .	518
8.474.2.9 state . . . . .	518
8.474.2.10userInputTimeout . . . . .	518
8.474.2.11version . . . . .	518
8.474.2.12versionlength . . . . .	518
8.475omaDmNotificationsTlv Struct Reference . . . . .	518
8.475.1 Field Documentation . . . . .	518
8.475.1.1 notification . . . . .	518
8.475.1.2 sessionStatus . . . . .	518
8.476operatorNameString Struct Reference . . . . .	518
8.476.1 Detailed Description . . . . .	519
8.476.2 Field Documentation . . . . .	519
8.476.2.1 PLMNName . . . . .	519
8.477OperatorPLMNData Struct Reference . . . . .	519
8.477.1 Detailed Description . . . . .	519
8.477.2 Field Documentation . . . . .	519
8.477.2.1 lac1 . . . . .	519
8.477.2.2 lac2 . . . . .	520

8.477.2.3 mcc . . . . .	520
8.477.2.4 mnc . . . . .	520
8.477.2.5 PLMNRecID . . . . .	520
8.478operatorPLMNList Struct Reference . . . . .	520
8.478.1 Detailed Description . . . . .	520
8.478.2 Field Documentation . . . . .	520
8.478.2.1 numInstance . . . . .	520
8.478.2.2 PLMNData . . . . .	520
8.479pack_dms_GetCustFeaturesV2_t Struct Reference . . . . .	520
8.479.1 Detailed Description . . . . .	520
8.479.2 Field Documentation . . . . .	521
8.479.2.1 cust_id . . . . .	521
8.479.2.2 list_type . . . . .	521
8.479.2.3 Tlvresult . . . . .	521
8.480pack_dms_SetCrashAction_t Struct Reference . . . . .	521
8.480.1 Detailed Description . . . . .	521
8.480.2 Field Documentation . . . . .	521
8.480.2.1 crashAction . . . . .	521
8.481pack_dms_SetCustFeature_t Struct Reference . . . . .	521
8.481.1 Field Documentation . . . . .	522
8.481.1.1 DHCPRelayEnabled . . . . .	522
8.481.1.2 DisableIMSI . . . . .	522
8.481.1.3 GpsEnable . . . . .	522
8.481.1.4 GPSLPM . . . . .	522
8.481.1.5 GPSSel . . . . .	522
8.481.1.6 IPFamSupport . . . . .	522
8.481.1.7 IsVoiceEnabled . . . . .	522
8.481.1.8 RMAutoConnect . . . . .	522
8.481.1.9 SMSSupport . . . . .	522
8.482pack_dms_SetCustFeaturesV2_t Struct Reference . . . . .	522
8.482.1 Detailed Description . . . . .	522
8.482.2 Field Documentation . . . . .	522
8.482.2.1 cust_id . . . . .	522
8.482.2.2 cust_value . . . . .	522
8.482.2.3 Tlvresult . . . . .	523
8.482.2.4 value_length . . . . .	523
8.483pack_dms_SetEventReport_t Struct Reference . . . . .	523
8.483.1 Field Documentation . . . . .	523
8.483.1.1 mode . . . . .	523
8.484pack_dms_SetPower_t Struct Reference . . . . .	523

8.484.1 Field Documentation . . . . .	523
8.484.1.1 mode . . . . .	523
8.484.1.2 Tlvresult . . . . .	523
8.485pack_dms_SetUSBComp_t Struct Reference . . . . .	523
8.485.1 Field Documentation . . . . .	523
8.485.1.1 Tlvresult . . . . .	523
8.485.1.2 USBComp . . . . .	523
8.486pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . . .	523
8.486.1 Detailed Description . . . . .	523
8.486.2 Field Documentation . . . . .	524
8.486.2.1 resetInfoInd . . . . .	524
8.487pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . . .	524
8.487.1 Detailed Description . . . . .	524
8.487.2 Field Documentation . . . . .	524
8.487.2.1 pDestSMSContent . . . . .	524
8.487.2.2 pDestSMSNum . . . . .	524
8.488pack_dms_UIMGetICCID_t Struct Reference . . . . .	524
8.488.1 Detailed Description . . . . .	524
8.488.2 Field Documentation . . . . .	525
8.488.2.1 Tlvresult . . . . .	525
8.489pack_fms_GetImagesPreference_t Struct Reference . . . . .	525
8.489.1 Detailed Description . . . . .	525
8.489.2 Field Documentation . . . . .	525
8.489.2.1 Tlvresult . . . . .	525
8.490pack_fms_GetStoredImages_t Struct Reference . . . . .	525
8.490.1 Detailed Description . . . . .	525
8.490.2 Field Documentation . . . . .	525
8.490.2.1 Tlvresult . . . . .	525
8.491pack_fms_SetImagesPreference_t Struct Reference . . . . .	525
8.491.1 Detailed Description . . . . .	526
8.491.2 Field Documentation . . . . .	526
8.491.2.1 bForceDownload . . . . .	526
8.491.2.2 imageListSize . . . . .	526
8.491.2.3 modemindex . . . . .	526
8.491.2.4 pImageList . . . . .	526
8.491.2.5 Tlvresult . . . . .	526
8.492pack_loc_Delete_Assist_Data_t Struct Reference . . . . .	526
8.492.1 Detailed Description . . . . .	526
8.492.2 Field Documentation . . . . .	527
8.492.2.1 pBdsSVInfo . . . . .	527

8.492.2.2 pCellDb . . . . .	527
8.492.2.3 pCikInfo . . . . .	527
8.492.2.4 pGnssData . . . . .	527
8.492.2.5 pSVInfo . . . . .	527
8.492.2.6 Tlvresult . . . . .	527
8.493pack_loc_EventRegister_t Struct Reference . . . . .	527
8.493.1 Detailed Description . . . . .	527
8.493.2 Field Documentation . . . . .	529
8.493.2.1 eventRegister . . . . .	529
8.493.2.2 Tlvresult . . . . .	529
8.494pack_loc_SetExtPowerState_t Struct Reference . . . . .	529
8.494.1 Detailed Description . . . . .	529
8.494.2 Field Documentation . . . . .	529
8.494.2.1 extPowerState . . . . .	529
8.494.2.2 Tlvresult . . . . .	529
8.495pack_loc_SetOperationMode_t Struct Reference . . . . .	529
8.495.1 Detailed Description . . . . .	530
8.495.2 Field Documentation . . . . .	530
8.495.2.1 mode . . . . .	530
8.495.2.2 Tlvresult . . . . .	530
8.496pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . . .	530
8.496.1 Detailed Description . . . . .	530
8.496.2 Field Documentation . . . . .	530
8.496.2.1 Tlvresult . . . . .	530
8.496.2.2 xid . . . . .	530
8.497pack_loc_Start_t Struct Reference . . . . .	530
8.497.1 Detailed Description . . . . .	531
8.497.2 Field Documentation . . . . .	532
8.497.2.1 pApplicationInfo . . . . .	532
8.497.2.2 pConfigAltitudeAssumed . . . . .	532
8.497.2.3 pHorizontalAccuracyLvl . . . . .	532
8.497.2.4 pIntermediateReportState . . . . .	532
8.497.2.5 pMinIntervalTime . . . . .	532
8.497.2.6 pRecurrenceType . . . . .	532
8.497.2.7 SessionId . . . . .	532
8.497.2.8 Tlvresult . . . . .	532
8.498pack_loc_Stop_t Struct Reference . . . . .	532
8.498.1 Detailed Description . . . . .	532
8.498.2 Field Documentation . . . . .	532
8.498.2.1 SessionId . . . . .	532

8.498.2.2 Tlvresult . . . . .	532
8.499pack_nas_SetACCOLC_t Struct Reference . . . . .	532
8.499.1 Detailed Description . . . . .	533
8.499.2 Field Documentation . . . . .	533
8.499.2.1 accolc . . . . .	533
8.499.2.2 spc . . . . .	533
8.500pack_nas_SetNetworkPreference_t Struct Reference . . . . .	533
8.500.1 Detailed Description . . . . .	533
8.500.2 Field Documentation . . . . .	533
8.500.2.1 Duration . . . . .	534
8.500.2.2 TechnologyPref . . . . .	534
8.500.2.3 Tlvresult . . . . .	534
8.501pack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	534
8.501.1 Detailed Description . . . . .	534
8.501.2 Field Documentation . . . . .	534
8.501.2.1 mcc . . . . .	534
8.501.2.2 mnc . . . . .	534
8.501.2.3 pMncPcsStatus . . . . .	534
8.502pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . . .	534
8.502.1 Detailed Description . . . . .	535
8.502.2 Field Documentation . . . . .	535
8.502.2.1 pChangeDuration . . . . .	535
8.502.2.2 pMncPcsDigitStatus . . . . .	535
8.502.2.3 pMNRInfo . . . . .	535
8.502.2.4 regAction . . . . .	535
8.503pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . . .	535
8.503.1 Detailed Description . . . . .	536
8.503.2 Field Documentation . . . . .	538
8.503.2.1 pCDMAECIODelta . . . . .	539
8.503.2.2 pCDMAECIOThresh . . . . .	539
8.503.2.3 pCDMARSSIDelta . . . . .	539
8.503.2.4 pCDMARSSIThresh . . . . .	539
8.503.2.5 pGSMRSSIDelta . . . . .	539
8.503.2.6 pGSMRSSIThresh . . . . .	539
8.503.2.7 pHDRECIODelta . . . . .	539
8.503.2.8 pHDRECIOThresh . . . . .	539
8.503.2.9 pHDRIDelta . . . . .	539
8.503.2.10pHDRIOThresh . . . . .	539
8.503.2.11pHDRRSSIDelta . . . . .	539
8.503.2.12pHDRRSSIThresh . . . . .	539

8.503.2.13	pHDRSINRDelta	539
8.503.2.14	pHDRSINRThresh	539
8.503.2.15	pLTERSRPDelta	539
8.503.2.16	pLTERSRPThresh	539
8.503.2.17	pLTERSQRDelta	539
8.503.2.18	pLTERSQRThresh	539
8.503.2.19	pLTERSSIDelta	539
8.503.2.20	pLTERSSIThresh	539
8.503.2.21	pLTESigRptConfig	539
8.503.2.22	pLTESNRDelta	539
8.503.2.23	pLTESNRThresh	539
8.503.2.24	pTDSCDMAECIODelta	539
8.503.2.25	pTDSCDMAECIOThresh	539
8.503.2.26	pTDSCDMARSCPDelta	539
8.503.2.27	pTDSCDMARSCPThresh	539
8.503.2.28	pTDSCDMARSSIDelta	539
8.503.2.29	pTDSCDMARSSIThresh	540
8.503.2.30	pTDSCDMASINRDelta	540
8.503.2.31	pTDSCDMASINRThresh	540
8.503.2.32	pWCDMAECIODelta	540
8.503.2.33	pWCDMAECIOThresh	540
8.503.2.34	pWCDMARSSIDelta	540
8.503.2.35	pWCDMARSSIThresh	540
8.504	pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference	540
8.504.1	Detailed Description	540
8.504.2	Field Documentation	542
8.504.2.1	pDDTMInd	542
8.504.2.2	pDualStandByPrefInd	542
8.504.2.3	pErrorRateInd	542
8.504.2.4	pHDRNewUATIAssInd	542
8.504.2.5	pHDRSessionCloseInd	542
8.504.2.6	pLTECphyCa	542
8.504.2.7	pManagedRoamingInd	542
8.504.2.8	pNetworkTimeInd	542
8.504.2.9	pServingSystemInd	542
8.504.2.10	pSignalStrengthInd	542
8.504.2.11	pSubscriptionInfoInd	542
8.504.2.12	pSysInfoInd	543
8.504.2.13	pSystemSelectionInd	543
8.505	pack_nas_SLQSNasSwiOTAMessageCallback_t Struct Reference	543

8.505.1 Detailed Description . . . . .	543
8.505.2 Field Documentation . . . . .	543
8.505.2.1 gsmUmtsDI . . . . .	544
8.505.2.2 gsmUmtsUI . . . . .	544
8.505.2.3 lteEmmDI . . . . .	544
8.505.2.4 lteEmmUI . . . . .	544
8.505.2.5 lteEsmDI . . . . .	544
8.505.2.6 lteEsmUI . . . . .	544
8.505.2.7 pRankIndicatorInd . . . . .	544
8.506pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference . . . . .	544
8.506.1 Detailed Description . . . . .	544
8.506.2 Field Documentation . . . . .	544
8.506.2.1 bEnable . . . . .	544
8.506.2.2 pSigIndReq . . . . .	544
8.507pack_nas_SLQSSetSysSelectionPref_t Struct Reference . . . . .	544
8.507.1 Detailed Description . . . . .	545
8.507.2 Field Documentation . . . . .	548
8.507.2.1 pAcqOrderPref . . . . .	548
8.507.2.2 pBandPref . . . . .	548
8.507.2.3 pChgDuration . . . . .	548
8.507.2.4 pCSGID . . . . .	548
8.507.2.5 pEmerMode . . . . .	548
8.507.2.6 pGWAcqOrderPref . . . . .	548
8.507.2.7 pLTEBandPref . . . . .	549
8.507.2.8 pMNCIncPCSDigStat . . . . .	549
8.507.2.9 pModePref . . . . .	549
8.507.2.10pNetSelPref . . . . .	549
8.507.2.11pPRLPref . . . . .	549
8.507.2.12pRAT . . . . .	549
8.507.2.13pRoamPref . . . . .	549
8.507.2.14pSrvDomainPref . . . . .	549
8.507.2.15pSrvRegRestriction . . . . .	549
8.507.2.16pTdsdmaBandPref . . . . .	549
8.508pack_qmi_t Struct Reference . . . . .	549
8.508.1 Detailed Description . . . . .	549
8.508.2 Field Documentation . . . . .	549
8.508.2.1 msgid . . . . .	549
8.508.2.2 svc . . . . .	549
8.508.2.3 timeout . . . . .	549
8.508.2.4 xid . . . . .	549

8.509	<a href="#">pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference</a>	549
8.509.1	Detailed Description	550
8.509.2	Field Documentation	550
8.509.2.1	apnId	550
8.510	<a href="#">pack_qos_SLQSQosSwiReadDataStats_t Struct Reference</a>	550
8.510.1	Detailed Description	550
8.510.2	Field Documentation	550
8.510.2.1	apnId	550
8.511	<a href="#">pack_qos_SLQSSetQosEventCallback_t Struct Reference</a>	550
8.511.1	Detailed Description	550
8.511.2	Field Documentation	551
8.511.2.1	enable	551
8.512	<a href="#">pack_sms_SendSMS_t Struct Reference</a>	551
8.512.1	Detailed Description	551
8.512.2	Field Documentation	551
8.512.2.1	messageFormat	551
8.512.2.2	messageSize	551
8.512.2.3	pLinktimer	551
8.512.2.4	pMessage	551
8.513	<a href="#">pack_sms_SetNewSMSCallback_t Struct Reference</a>	551
8.513.1	Detailed Description	552
8.513.2	Field Documentation	552
8.513.2.1	status	552
8.514	<a href="#">pack_sms_SLQSDDeleteSMS_t Struct Reference</a>	552
8.514.1	Detailed Description	552
8.514.2	Field Documentation	552
8.514.2.1	pMessageIndex	553
8.514.2.2	pMessageMode	553
8.514.2.3	pMessageTag	553
8.514.2.4	storageType	553
8.515	<a href="#">pack_sms_SLQSGetSMS_t Struct Reference</a>	553
8.515.1	Detailed Description	553
8.515.2	Field Documentation	553
8.515.2.1	messageIndex	553
8.515.2.2	pMessageMode	553
8.515.2.3	storageType	553
8.516	<a href="#">pack_sms_SLQSGetSMSList_t Struct Reference</a>	553
8.516.1	Detailed Description	553
8.516.2	Field Documentation	554
8.516.2.1	pMessageMode	554

8.516.2.2 pRequestedTag . . . . .	554
8.516.2.3 storageType . . . . .	554
8.517pack_sms_SLQSMModifySMSStatus_t Struct Reference . . . . .	554
8.517.1 Detailed Description . . . . .	554
8.517.2 Field Documentation . . . . .	555
8.517.2.1 messageIndex . . . . .	555
8.517.2.2 messageTag . . . . .	555
8.517.2.3 pMessageMode . . . . .	555
8.517.2.4 storageType . . . . .	555
8.518pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . . .	555
8.518.1 Detailed Description . . . . .	555
8.518.2 Field Documentation . . . . .	556
8.518.2.1 fix_rate . . . . .	556
8.518.2.2 fix_type . . . . .	556
8.518.2.3 function . . . . .	556
8.518.2.4 max_dist . . . . .	556
8.518.2.5 max_time . . . . .	556
8.518.2.6 set_fix_rate . . . . .	556
8.518.2.7 set_fix_type . . . . .	556
8.518.2.8 set_function . . . . .	556
8.518.2.9 set_max_dist . . . . .	556
8.518.2.10set_max_time . . . . .	556
8.519pack_swioama_SLQSOMADMCancelSession_t Struct Reference . . . . .	556
8.519.1 Detailed Description . . . . .	557
8.519.2 Field Documentation . . . . .	557
8.519.2.1 sessionType . . . . .	557
8.520pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	557
8.520.1 Detailed Description . . . . .	557
8.520.2 Field Documentation . . . . .	557
8.520.2.1 SessionType . . . . .	557
8.521pack_swioama_SLQSOMADMSelectSendSelection_t Struct Reference . . . . .	557
8.521.1 Detailed Description . . . . .	558
8.521.2 Field Documentation . . . . .	558
8.521.2.1 pDeferTime . . . . .	558
8.521.2.2 pRejectReason . . . . .	558
8.521.2.3 selection . . . . .	558
8.522pack_swioama_SLQSOMADMSetSettings_t Struct Reference . . . . .	558
8.522.1 Detailed Description . . . . .	558
8.522.2 Field Documentation . . . . .	559
8.522.2.1 FOTAdownload . . . . .	559

8.522.2.2 FOTAUpdate . . . . .	559
8.522.2.3 pAutosdm . . . . .	559
8.522.2.4 pFwAutoCheck . . . . .	559
8.523pack_swima_SLQSOMADMStartSession_t Struct Reference . . . . .	559
8.523.1 Detailed Description . . . . .	559
8.523.2 Field Documentation . . . . .	559
8.523.2.1 sessionType . . . . .	559
8.524pack_uim_ChangePin_t Struct Reference . . . . .	560
8.524.1 Detailed Description . . . . .	560
8.524.2 Field Documentation . . . . .	560
8.524.2.1 changePIN . . . . .	560
8.524.2.2 EncryptedPIN1 . . . . .	560
8.524.2.3 pIndicationToken . . . . .	560
8.524.2.4 pKeyReferenceID . . . . .	560
8.524.2.5 sessionInfo . . . . .	560
8.524.2.6 Tlvresult . . . . .	561
8.525pack_uim_ReadTransparent_t Struct Reference . . . . .	561
8.525.1 Detailed Description . . . . .	561
8.525.2 Field Documentation . . . . .	561
8.525.2.1 fileIndex . . . . .	561
8.525.2.2 pEncryptData . . . . .	561
8.525.2.3 pIndicationToken . . . . .	561
8.525.2.4 readTransparent . . . . .	562
8.525.2.5 sessionInfo . . . . .	562
8.525.2.6 Tlvresult . . . . .	562
8.526pack_uim_SetPinProtection_t Struct Reference . . . . .	562
8.526.1 Detailed Description . . . . .	562
8.526.2 Field Documentation . . . . .	562
8.526.2.1 EncryptedPIN1 . . . . .	562
8.526.2.2 pIndicationToken . . . . .	562
8.526.2.3 pinProtection . . . . .	563
8.526.2.4 pKeyReferenceID . . . . .	563
8.526.2.5 sessionInfo . . . . .	563
8.526.2.6 Tlvresult . . . . .	563
8.527pack_uim_SLQSUIEventRegister_t Struct Reference . . . . .	563
8.527.1 Detailed Description . . . . .	563
8.527.2 Field Documentation . . . . .	563
8.527.2.1 eventMask . . . . .	563
8.528pack_uim_SLQSUIMSwitchSlot_t Struct Reference . . . . .	563
8.528.1 Detailed Description . . . . .	563

8.528.2 Field Documentation . . . . .	564
8.528.2.1 bLogicalSlot . . . . .	564
8.528.2.2 ulPhysicalSlot . . . . .	564
8.529pack_uim_UnblockPin_t Struct Reference . . . . .	564
8.529.1 Detailed Description . . . . .	564
8.529.2 Field Documentation . . . . .	565
8.529.2.1 EncryptedPIN1 . . . . .	565
8.529.2.2 pIndicationToken . . . . .	565
8.529.2.3 pinProtection . . . . .	565
8.529.2.4 pKeyReferenceID . . . . .	565
8.529.2.5 sessionInfo . . . . .	565
8.529.2.6 Tlvresult . . . . .	565
8.530pack_uim_VerifyPin_t Struct Reference . . . . .	565
8.530.1 Detailed Description . . . . .	565
8.530.2 Field Documentation . . . . .	566
8.530.2.1 pEncryptedPIN1 . . . . .	566
8.530.2.2 pIndicationToken . . . . .	566
8.530.2.3 pKeyReferenceID . . . . .	566
8.530.2.4 sessionInfo . . . . .	566
8.530.2.5 Tlvresult . . . . .	566
8.530.2.6 verifyPIN . . . . .	566
8.531pack_wds_GetDefaultProfile_t Struct Reference . . . . .	566
8.531.1 Detailed Description . . . . .	566
8.531.2 Field Documentation . . . . .	566
8.531.2.1 profiletype . . . . .	566
8.532pack_wds_GetDefaultProfileNum_t Struct Reference . . . . .	566
8.532.1 Detailed Description . . . . .	567
8.532.2 Field Documentation . . . . .	567
8.532.2.1 family . . . . .	567
8.532.2.2 type . . . . .	567
8.533pack_wds_GetDormancyState_t Struct Reference . . . . .	567
8.534pack_wds_GetLastMobileIPError_t Struct Reference . . . . .	567
8.535pack_wds_GetMobileIP_t Struct Reference . . . . .	567
8.536pack_wds_GetMobileIPProfile_t Struct Reference . . . . .	567
8.536.1 Detailed Description . . . . .	567
8.536.2 Field Documentation . . . . .	567
8.536.2.1 index . . . . .	567
8.537pack_wds_GetPacketStatus_t Struct Reference . . . . .	567
8.537.1 Detailed Description . . . . .	568
8.537.2 Field Documentation . . . . .	568

8.537.2.1 statmask . . . . .	568
8.538pack_wds_GetSessionDuration_t Struct Reference . . . . .	568
8.539pack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	568
8.539.1 Detailed Description . . . . .	568
8.539.2 Field Documentation . . . . .	568
8.539.2.1 RmTrasnferStaticsReq . . . . .	568
8.540pack_wds_SetDefaultProfile_t Struct Reference . . . . .	568
8.540.1 Detailed Description . . . . .	568
8.540.2 Field Documentation . . . . .	569
8.540.2.1 authentication . . . . .	569
8.540.2.2 ipAddress . . . . .	569
8.540.2.3 pApnname . . . . .	569
8.540.2.4 pdpType . . . . .	569
8.540.2.5 pName . . . . .	569
8.540.2.6 pPassword . . . . .	569
8.540.2.7 primaryDNS . . . . .	569
8.540.2.8 profileType . . . . .	569
8.540.2.9 pUsername . . . . .	569
8.540.2.10secondaryDNS . . . . .	569
8.541pack_wds_SetDefaultProfileNum_t Struct Reference . . . . .	569
8.541.1 Field Documentation . . . . .	569
8.541.1.1 family . . . . .	569
8.541.1.2 index . . . . .	569
8.541.1.3 type . . . . .	569
8.542pack_wds_SetMobileIPProfile_t Struct Reference . . . . .	569
8.542.1 Detailed Description . . . . .	570
8.542.2 Field Documentation . . . . .	570
8.542.2.1 index . . . . .	570
8.542.2.2 pAAASPI . . . . .	570
8.542.2.3 pAddress . . . . .	570
8.542.2.4 pEnabled . . . . .	570
8.542.2.5 pHASPI . . . . .	570
8.542.2.6 pMNAAA . . . . .	570
8.542.2.7 pMNHA . . . . .	570
8.542.2.8 pNAI . . . . .	570
8.542.2.9 pPrimaryHA . . . . .	570
8.542.2.10pRevTunneling . . . . .	570
8.542.2.11pSecondaryHA . . . . .	571
8.542.2.12spc . . . . .	571
8.543pack_wds_SLQSCreateProfile_t Struct Reference . . . . .	571

8.543.1 Detailed Description . . . . .	571
8.543.2 Field Documentation . . . . .	571
8.543.2.1 pCurProfile . . . . .	571
8.543.2.2 pProfileId . . . . .	571
8.543.2.3 pProfileType . . . . .	571
8.544pack_wds_SLQSDeleteProfile_t Struct Reference . . . . .	571
8.544.1 Detailed Description . . . . .	571
8.544.2 Field Documentation . . . . .	572
8.544.2.1 profileIndex . . . . .	572
8.544.2.2 profileType . . . . .	572
8.545pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	572
8.546pack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . . .	572
8.547pack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . . .	572
8.547.1 Detailed Description . . . . .	572
8.547.2 Field Documentation . . . . .	572
8.547.2.1 Mask . . . . .	572
8.547.2.2 pReportChannelRate . . . . .	572
8.547.2.3 pReportConnStatus . . . . .	572
8.547.2.4 pReportDataBearerTech . . . . .	572
8.547.2.5 pReportDormStatus . . . . .	572
8.547.2.6 pTransferStatInd . . . . .	573
8.548pack_wds_SLQSGetProfileSettings_t Struct Reference . . . . .	573
8.548.1 Detailed Description . . . . .	573
8.548.2 Field Documentation . . . . .	573
8.548.2.1 ProfileId . . . . .	573
8.548.2.2 ProfileType . . . . .	573
8.549pack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . . .	573
8.549.1 Detailed Description . . . . .	573
8.549.2 Field Documentation . . . . .	574
8.549.2.1 pReqSettings . . . . .	574
8.550pack_wds_SLQSModifyProfile_t Struct Reference . . . . .	574
8.550.1 Detailed Description . . . . .	574
8.550.2 Field Documentation . . . . .	575
8.550.2.1 curProfile . . . . .	575
8.550.2.2 pProfileId . . . . .	575
8.550.2.3 pProfileType . . . . .	575
8.551pack_wds_SLQSSet3GPPConfigItem_t Struct Reference . . . . .	575
8.551.1 Detailed Description . . . . .	575
8.551.2 Field Documentation . . . . .	576
8.551.2.1 _3gppRelease . . . . .	576

8.551.2.2 defaultPDNEnabled . . . . .	576
8.551.2.3 LTEAttachProfileList . . . . .	576
8.551.2.4 LTEAttachProfileListLen . . . . .	576
8.551.2.5 profileList . . . . .	576
8.552pack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . . .	576
8.552.1 Detailed Description . . . . .	576
8.552.2 Field Documentation . . . . .	576
8.552.2.1 IPFamilyPreference . . . . .	576
8.553pack_wds_SLQSSetWdsEventCallback_t Struct Reference . . . . .	576
8.553.1 Detailed Description . . . . .	577
8.553.2 Field Documentation . . . . .	577
8.553.2.1 currentDataBearer . . . . .	577
8.553.2.2 dataBearer . . . . .	577
8.553.2.3 dataSystemStatus . . . . .	577
8.553.2.4 dormancyStatus . . . . .	577
8.553.2.5 interval . . . . .	577
8.553.2.6 mobileIP . . . . .	577
8.553.2.7 transferStats . . . . .	577
8.554pack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference . . . . .	577
8.554.1 Detailed Description . . . . .	577
8.554.2 Field Documentation . . . . .	577
8.554.2.1 pProfileId . . . . .	577
8.555pack_wds_SLQSStartDataSession_t Struct Reference . . . . .	577
8.555.1 Detailed Description . . . . .	578
8.555.2 Field Documentation . . . . .	578
8.555.2.1 pAuth . . . . .	578
8.555.2.2 pPass . . . . .	578
8.555.2.3 pprofileid3gpp . . . . .	578
8.555.2.4 pprofileid3gpp2 . . . . .	578
8.555.2.5 pTech . . . . .	579
8.555.2.6 pUser . . . . .	579
8.556pack_wds_SLQSStopDataSession_t Struct Reference . . . . .	579
8.556.1 Detailed Description . . . . .	579
8.556.2 Field Documentation . . . . .	579
8.556.2.1 psid . . . . .	579
8.557pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . . .	579
8.557.1 Detailed Description . . . . .	579
8.557.2 Field Documentation . . . . .	579
8.557.2.1 contextId . . . . .	579
8.557.2.2 contextType . . . . .	579

8.558PackCreateProfileOut Struct Reference . . . . .	579
8.558.1 Field Documentation . . . . .	579
8.558.1.1 ExtErrorCode . . . . .	579
8.558.1.2 ProfileIndex . . . . .	580
8.558.1.3 ProfileType . . . . .	580
8.559packgetDyingGaspCfg Struct Reference . . . . .	580
8.559.1 Detailed Description . . . . .	580
8.559.2 Field Documentation . . . . .	580
8.559.2.1 pDestSMSContent . . . . .	580
8.559.2.2 pDestSMSNum . . . . .	580
8.560packgetDyingGaspStatistics Struct Reference . . . . .	580
8.560.1 Detailed Description . . . . .	580
8.560.2 Field Documentation . . . . .	580
8.560.2.1 pSMSAttemptedFlag . . . . .	580
8.560.2.2 pTimeStamp . . . . .	581
8.561PCMparams Struct Reference . . . . .	581
8.561.1 Detailed Description . . . . .	581
8.561.2 Field Documentation . . . . .	581
8.561.2.1 iFaceTab . . . . .	581
8.561.2.2 iFaceTabLen . . . . .	581
8.562PCSCFFQDNAddress Struct Reference . . . . .	581
8.562.1 Detailed Description . . . . .	581
8.562.2 Field Documentation . . . . .	581
8.562.2.1 fqdnAddr . . . . .	582
8.562.2.2 fqdnLen . . . . .	582
8.563PCSCFFQDNAddressList Struct Reference . . . . .	582
8.563.1 Detailed Description . . . . .	582
8.563.2 Field Documentation . . . . .	582
8.563.2.1 numInstances . . . . .	582
8.563.2.2 pcsfFQDNAddress . . . . .	582
8.564PCSCFIPv4ServerAddressList Struct Reference . . . . .	582
8.564.1 Detailed Description . . . . .	582
8.564.2 Field Documentation . . . . .	582
8.564.2.1 numInstances . . . . .	583
8.564.2.2 pscsfIPv4Addr . . . . .	583
8.565PDSPositionData Struct Reference . . . . .	583
8.565.1 Detailed Description . . . . .	583
8.565.2 Field Documentation . . . . .	584
8.565.2.1 pAltitudeWrtEllipsoid . . . . .	584
8.565.2.2 pAltitudeWrtSealevel . . . . .	584

8.565.2.3 pHorizontalConfidence . . . . .	584
8.565.2.4 pHorizontalUncCircular . . . . .	584
8.565.2.5 pLatitude . . . . .	584
8.565.2.6 pLongitude . . . . .	584
8.565.2.7 pPositionSource . . . . .	584
8.565.2.8 pTimeStamp . . . . .	584
8.565.2.9 pTimeType . . . . .	584
8.565.2.10 pVerticalConfidence . . . . .	584
8.565.2.11 pVerticalUnc . . . . .	585
8.566 PDSPosMethodStateReq Struct Reference . . . . .	585
8.566.1 Detailed Description . . . . .	585
8.566.2 Field Documentation . . . . .	585
8.566.2.1 pWifiState . . . . .	585
8.566.2.2 pXtraDataState . . . . .	585
8.566.2.3 pXtraTimeState . . . . .	585
8.567 peerNumberInfo Struct Reference . . . . .	585
8.567.1 Detailed Description . . . . .	586
8.567.2 Field Documentation . . . . .	587
8.567.2.1 callID . . . . .	587
8.567.2.2 number . . . . .	587
8.567.2.3 numLen . . . . .	587
8.567.2.4 numPI . . . . .	587
8.567.2.5 numPlan . . . . .	587
8.567.2.6 numSI . . . . .	587
8.567.2.7 numType . . . . .	587
8.568 personalizationStatus Struct Reference . . . . .	587
8.568.1 Detailed Description . . . . .	587
8.568.2 Field Documentation . . . . .	588
8.568.2.1 feature . . . . .	588
8.568.2.2 numFeatures . . . . .	588
8.568.2.3 unblockLeft . . . . .	588
8.568.2.4 verifyLeft . . . . .	588
8.569 PhyCaAggPcellInfo Struct Reference . . . . .	588
8.569.1 Detailed Description . . . . .	588
8.569.2 Field Documentation . . . . .	589
8.569.2.1 dl_bw_value . . . . .	589
8.569.2.2 freq . . . . .	589
8.569.2.3 iLTEbandValue . . . . .	589
8.569.2.4 pci . . . . .	589
8.569.2.5 TlvPresent . . . . .	589

8.570PhyCaAggScellDIBw Struct Reference . . . . .	589
8.570.1 Detailed Description . . . . .	589
8.570.2 Field Documentation . . . . .	589
8.570.2.1 dl_bw_value . . . . .	590
8.570.2.2 TlvPresent . . . . .	590
8.571PhyCaAggScellIndex Struct Reference . . . . .	590
8.571.1 Detailed Description . . . . .	590
8.571.2 Field Documentation . . . . .	590
8.571.2.1 scell_idx . . . . .	590
8.571.2.2 TlvPresent . . . . .	590
8.572PhyCaAggScellIndType Struct Reference . . . . .	590
8.572.1 Detailed Description . . . . .	590
8.572.2 Field Documentation . . . . .	591
8.572.2.1 freq . . . . .	591
8.572.2.2 pci . . . . .	591
8.572.2.3 scell_state . . . . .	591
8.572.2.4 TlvPresent . . . . .	591
8.573PhyCaAggScellInfo Struct Reference . . . . .	591
8.573.1 Detailed Description . . . . .	591
8.573.2 Field Documentation . . . . .	593
8.573.2.1 dl_bw_value . . . . .	593
8.573.2.2 freq . . . . .	593
8.573.2.3 iLTEbandValue . . . . .	593
8.573.2.4 pci . . . . .	593
8.573.2.5 scell_state . . . . .	593
8.573.2.6 TlvPresent . . . . .	593
8.574PilotSetData Struct Reference . . . . .	593
8.574.1 Detailed Description . . . . .	593
8.574.2 Field Documentation . . . . .	593
8.574.2.1 NumPilots . . . . .	593
8.574.2.2 pPilotSetInfo . . . . .	593
8.575PilotSetParams Struct Reference . . . . .	594
8.575.1 Detailed Description . . . . .	594
8.575.2 Field Documentation . . . . .	594
8.575.2.1 PilotPN . . . . .	594
8.575.2.2 PilotStrength . . . . .	594
8.575.2.3 PilotType . . . . .	594
8.576pktErrRate Struct Reference . . . . .	594
8.576.1 Detailed Description . . . . .	594
8.576.2 Field Documentation . . . . .	594

8.576.2.1 exponent . . . . .	594
8.576.2.2 multiplier . . . . .	595
8.577PLMNNetworkName Struct Reference . . . . .	595
8.577.1 Detailed Description . . . . .	595
8.577.2 Field Documentation . . . . .	595
8.577.2.1 numInstance . . . . .	595
8.577.2.2 PLMNNetName . . . . .	595
8.578PLMNNetworkNameData Struct Reference . . . . .	595
8.578.1 Detailed Description . . . . .	595
8.578.2 Field Documentation . . . . .	596
8.578.2.1 codingScheme . . . . .	596
8.578.2.2 countryInitials . . . . .	596
8.578.2.3 longName . . . . .	596
8.578.2.4 longNameLen . . . . .	597
8.578.2.5 longNameSpareBits . . . . .	597
8.578.2.6 shortName . . . . .	597
8.578.2.7 shortNameLen . . . . .	597
8.578.2.8 shortNameSpareBits . . . . .	597
8.579Port Struct Reference . . . . .	597
8.579.1 Detailed Description . . . . .	597
8.579.2 Field Documentation . . . . .	597
8.579.2.1 port . . . . .	597
8.579.2.2 range . . . . .	597
8.580precisionDilution_s Struct Reference . . . . .	597
8.580.1 Detailed Description . . . . .	597
8.580.2 Field Documentation . . . . .	598
8.580.2.1 HDOP . . . . .	598
8.580.2.2 PDOP . . . . .	598
8.580.2.3 VDOP . . . . .	598
8.581PrefImageList Struct Reference . . . . .	598
8.581.1 Detailed Description . . . . .	598
8.581.2 Field Documentation . . . . .	598
8.581.2.1 listEntries . . . . .	598
8.581.2.2 listSize . . . . .	598
8.582prefVoiceSO Struct Reference . . . . .	598
8.582.1 Detailed Description . . . . .	599
8.582.2 Field Documentation . . . . .	600
8.582.2.1 evrcCapability . . . . .	600
8.582.2.2 homeOrigVoiceSO . . . . .	600
8.582.2.3 homePageVoiceSO . . . . .	600

8.582.2.4 namID . . . . .	600
8.582.2.5 roamOrigVoiceSO . . . . .	600
8.583Profile3GPP Struct Reference . . . . .	600
8.583.1 Detailed Description . . . . .	601
8.583.2 Field Documentation . . . . .	605
8.583.2.1 pAddrAllocPref . . . . .	605
8.583.2.2 pAPNClass . . . . .	605
8.583.2.3 pAPNDisabledFlag . . . . .	605
8.583.2.4 pAPNName . . . . .	605
8.583.2.5 pAPNnameSize . . . . .	605
8.583.2.6 pAuthenticationPref . . . . .	605
8.583.2.7 pGPRSMinimumQoS . . . . .	605
8.583.2.8 pGPRSRequestedQos . . . . .	605
8.583.2.9 plmCnFlag . . . . .	605
8.583.2.10pIPv4AddrPref . . . . .	605
8.583.2.11pIPv6AddPref . . . . .	605
8.583.2.12pPassword . . . . .	605
8.583.2.13pPasswordSize . . . . .	605
8.583.2.14pPcscfAddrUsingDhcp . . . . .	605
8.583.2.15pPcscfAddrUsingPCO . . . . .	605
8.583.2.16pPDNInactivTimeout . . . . .	605
8.583.2.17pPdpAccessConFlag . . . . .	605
8.583.2.18pPdpContext . . . . .	605
8.583.2.19pPdpDataCompType . . . . .	606
8.583.2.20pPdpHdrCompType . . . . .	606
8.583.2.21pPDType . . . . .	606
8.583.2.22pPriDNSIPv4AddPref . . . . .	606
8.583.2.23pPriDNSIPv6addpref . . . . .	606
8.583.2.24pPrimaryID . . . . .	606
8.583.2.25pProfilename . . . . .	606
8.583.2.26pProfilenameSize . . . . .	606
8.583.2.27pQosClassID . . . . .	606
8.583.2.28pSecDNSIPv4AddPref . . . . .	606
8.583.2.29pSecDNSIPv6addpref . . . . .	606
8.583.2.30pSecondaryFlag . . . . .	606
8.583.2.31pTFTID1Params . . . . .	606
8.583.2.32pTFTID2Params . . . . .	606
8.583.2.33pUMTSMinQoS . . . . .	606
8.583.2.34pUMTSMinQoSSigInd . . . . .	606
8.583.2.35pUMTSReqQoS . . . . .	606

8.583.2.36pUMTSReqQoSsigInd . . . . .	606
8.583.2.37pUsername . . . . .	606
8.583.2.38pUsernameSize . . . . .	606
8.584Profile3GPP2 Struct Reference . . . . .	606
8.584.1 Detailed Description . . . . .	607
8.584.2 Field Documentation . . . . .	610
8.584.2.1 pAllowLinger . . . . .	610
8.584.2.2 pAPNClass3GPP2 . . . . .	610
8.584.2.3 pAPNEnabled3GPP2 . . . . .	610
8.584.2.4 pApnString . . . . .	610
8.584.2.5 pApnStringSize . . . . .	610
8.584.2.6 pAppPriority . . . . .	610
8.584.2.7 pAppType . . . . .	610
8.584.2.8 pAuthPassword . . . . .	611
8.584.2.9 pAuthPasswordSize . . . . .	611
8.584.2.10pAuthProtocol . . . . .	611
8.584.2.11pAuthRetryCount . . . . .	611
8.584.2.12pAuthTimeout . . . . .	611
8.584.2.13pDataMode . . . . .	611
8.584.2.14pDataRate . . . . .	611
8.584.2.15pIpcpAckTimeout . . . . .	611
8.584.2.16pIpcpCreqRetryCount . . . . .	611
8.584.2.17pIsPcscfAddressNedded . . . . .	611
8.584.2.18pLcpAckTimeout . . . . .	611
8.584.2.19pLcpCreqRetryCount . . . . .	611
8.584.2.20pNegoDnsSrvrPref . . . . .	611
8.584.2.21pPDNInactivTimeout3GPP2 . . . . .	611
8.584.2.22pPdnType . . . . .	611
8.584.2.23pPppSessCloseTimer1x . . . . .	611
8.584.2.24pPppSessCloseTimerDO . . . . .	611
8.584.2.25pPrimaryV4DnsAddress . . . . .	611
8.584.2.26pPriV6DnsAddress . . . . .	611
8.584.2.27pRATType . . . . .	611
8.584.2.28pSecondaryV4DnsAddress . . . . .	611
8.584.2.29pSecV6DnsAddress . . . . .	611
8.584.2.30pUserId . . . . .	611
8.584.2.31pUserIdSize . . . . .	611
8.585ProfileIdentifier Struct Reference . . . . .	611
8.585.1 Detailed Description . . . . .	612
8.585.2 Field Documentation . . . . .	612

8.585.2.1 profileIndex . . . . .	612
8.585.2.2 profileType . . . . .	612
8.586protocolSubtypeElement Struct Reference . . . . .	612
8.586.1 Detailed Description . . . . .	612
8.586.2 Field Documentation . . . . .	613
8.586.2.1 AccessMac . . . . .	613
8.586.2.2 AuthProt . . . . .	613
8.586.2.3 ControlMac . . . . .	613
8.586.2.4 EncryptProt . . . . .	613
8.586.2.5 ForwardMac . . . . .	613
8.586.2.6 IdleState . . . . .	613
8.586.2.7 KeyExchange . . . . .	613
8.586.2.8 MultDisc . . . . .	613
8.586.2.9 PhysicalLayer . . . . .	613
8.586.2.10ReverseMac . . . . .	613
8.586.2.11SecProt . . . . .	614
8.586.2.12VirtStream . . . . .	614
8.587PSDetachReq Struct Reference . . . . .	614
8.587.1 Detailed Description . . . . .	614
8.587.2 Field Documentation . . . . .	614
8.587.2.1 pDetachAction . . . . .	614
8.588qaQmi3Gpp2TimeZone Struct Reference . . . . .	614
8.588.1 Detailed Description . . . . .	614
8.588.2 Field Documentation . . . . .	615
8.588.2.1 daylightSavings . . . . .	615
8.588.2.2 leapSeconds . . . . .	615
8.588.2.3 localTimeOffset . . . . .	615
8.589qaQmiInterfaceInfo Struct Reference . . . . .	615
8.589.1 Detailed Description . . . . .	615
8.589.2 Field Documentation . . . . .	615
8.589.2.1 qaQmiinstanceid . . . . .	615
8.589.2.2 qaQmisvctype . . . . .	615
8.589.2.3 v4sessionId . . . . .	615
8.589.2.4 v6sessionId . . . . .	615
8.590qaQmiServingSystemParam Struct Reference . . . . .	615
8.590.1 Detailed Description . . . . .	616
8.590.2 Field Documentation . . . . .	618
8.590.2.1 BasestationID . . . . .	618
8.590.2.2 BasestationLatitude . . . . .	618
8.590.2.3 BasestationLongitude . . . . .	618

8.590.2.4 CallBarStatus . . . . .	619
8.590.2.5 CDMA_P_Rev . . . . .	619
8.590.2.6 CDMASystemInfoExt . . . . .	619
8.590.2.7 CellID . . . . .	619
8.590.2.8 concSvcInfo . . . . .	619
8.590.2.9 CurrentPLMN . . . . .	619
8.590.2.10DataSrvCapabilities . . . . .	619
8.590.2.11defaultRoamInd . . . . .	619
8.590.2.12DetailedSvcInfo . . . . .	619
8.590.2.13DTMInd . . . . .	619
8.590.2.14Gpp2TimeZone . . . . .	619
8.590.2.15GppNetworkDSTAdjustment . . . . .	619
8.590.2.16GppTimeZone . . . . .	619
8.590.2.17hdrPersonality . . . . .	619
8.590.2.18Lac . . . . .	619
8.590.2.19NetworkID . . . . .	619
8.590.2.20PRLInd . . . . .	619
8.590.2.21roamIndicatorVal . . . . .	619
8.590.2.22RoamingIndicatorList . . . . .	619
8.590.2.23ServingSystem . . . . .	619
8.590.2.24SystemID . . . . .	619
8.590.2.25TrackAreaCode . . . . .	619
8.591QmiCbkCatEventStatusReportInd Struct Reference . . . . .	619
8.591.1 Field Documentation . . . . .	619
8.591.1.1 CCETiv . . . . .	619
8.591.1.2 event_Index . . . . .	620
8.592QmiCbkLocBestAvailPosInd Struct Reference . . . . .	620
8.592.1 Detailed Description . . . . .	620
8.592.2 Field Documentation . . . . .	624
8.592.2.1 pAltitudeWrtEllipsoid . . . . .	624
8.592.2.2 pAltitudeWrtMeanSeaLevel . . . . .	624
8.592.2.3 pGpsTime . . . . .	624
8.592.2.4 pHeading . . . . .	624
8.592.2.5 pHeadingUnc . . . . .	624
8.592.2.6 pHorCirConf . . . . .	624
8.592.2.7 pHorEllpConf . . . . .	624
8.592.2.8 pHorReliability . . . . .	624
8.592.2.9 pHorUncCircular . . . . .	624
8.592.2.10pHorUncEllipseOrientAzimuth . . . . .	624
8.592.2.11pHorUncEllipseSemiMajor . . . . .	624

8.592.2.12pHorUncEllipseSemiMinor . . . . .	625
8.592.2.13pLatitude . . . . .	625
8.592.2.14pLongitude . . . . .	625
8.592.2.15pMagneticDeviation . . . . .	625
8.592.2.16pPrecisionDilution . . . . .	625
8.592.2.17pSensorDataUsage . . . . .	625
8.592.2.18pSpeedHorizontal . . . . .	625
8.592.2.19pSpeedUnc . . . . .	625
8.592.2.20pSpeedVertical . . . . .	625
8.592.2.21pSpeedVerticalUnc . . . . .	625
8.592.2.22pSvUsedforFix . . . . .	625
8.592.2.23pTechnologyMask . . . . .	625
8.592.2.24pTimeSrc . . . . .	625
8.592.2.25pTimestampUtc . . . . .	625
8.592.2.26pTimeUnc . . . . .	625
8.592.2.27pVertConfidence . . . . .	625
8.592.2.28pVertReliability . . . . .	625
8.592.2.29pVertUnc . . . . .	625
8.592.2.30pXid . . . . .	625
8.592.2.31status . . . . .	625
8.593QmiCbkLocCradleMountInd Struct Reference . . . . .	625
8.593.1 Detailed Description . . . . .	625
8.593.2 Field Documentation . . . . .	626
8.593.2.1 cradleMountConfigStatus . . . . .	626
8.594QmiCbkLocEngineStateInd Struct Reference . . . . .	626
8.594.1 Detailed Description . . . . .	626
8.594.2 Field Documentation . . . . .	626
8.594.2.1 engineState . . . . .	626
8.595QmiCbkLocEventTimeSyncInd Struct Reference . . . . .	626
8.595.1 Detailed Description . . . . .	627
8.595.2 Field Documentation . . . . .	627
8.595.2.1 timeSyncRefCounter . . . . .	627
8.596QmiCbkLocInjectPositionInd Struct Reference . . . . .	627
8.596.1 Detailed Description . . . . .	627
8.596.2 Field Documentation . . . . .	627
8.596.2.1 status . . . . .	628
8.597QmiCbkLocInjectSensorDataInd Struct Reference . . . . .	628
8.597.1 Detailed Description . . . . .	628
8.597.2 Field Documentation . . . . .	629
8.597.2.1 injectSensorDataStatus . . . . .	629

8.597.2.2 pAccelSamplesAccepted . . . . .	629
8.597.2.3 pAccelTempSamplesAccepted . . . . .	629
8.597.2.4 pGyroSamplesAccepted . . . . .	629
8.597.2.5 pGyroTempSamplesAccepted . . . . .	629
8.597.2.6 pOpaqueIdentifier . . . . .	629
8.598QmiCbkLocInjectTimeInd Struct Reference . . . . .	629
8.598.1 Detailed Description . . . . .	629
8.598.2 Field Documentation . . . . .	629
8.598.2.1 injectTimeSyncStatus . . . . .	629
8.599QmiCbkLocInjectUTCTimeInd Struct Reference . . . . .	629
8.599.1 Detailed Description . . . . .	630
8.599.2 Field Documentation . . . . .	630
8.599.2.1 status . . . . .	630
8.600QmiCbkLocPositionReportInd Struct Reference . . . . .	630
8.600.1 Detailed Description . . . . .	631
8.600.2 Field Documentation . . . . .	635
8.600.2.1 pAltitudeAssumed . . . . .	635
8.600.2.2 pAltitudeWrtEllipsoid . . . . .	635
8.600.2.3 pAltitudeWrtMeanSeaLevel . . . . .	635
8.600.2.4 pFixId . . . . .	635
8.600.2.5 pGpsTime . . . . .	635
8.600.2.6 pHeading . . . . .	635
8.600.2.7 pHeadingUnc . . . . .	635
8.600.2.8 pHorConfidence . . . . .	635
8.600.2.9 pHorReliability . . . . .	635
8.600.2.10pHorUncCircular . . . . .	635
8.600.2.11pHorUncEllipseOrientAzimuth . . . . .	635
8.600.2.12pHorUncEllipseSemiMajor . . . . .	635
8.600.2.13pHorUncEllipseSemiMinor . . . . .	635
8.600.2.14pLatitude . . . . .	635
8.600.2.15pLeapSeconds . . . . .	635
8.600.2.16pLongitude . . . . .	635
8.600.2.17pMagneticDeviation . . . . .	635
8.600.2.18pPrecisionDilution . . . . .	635
8.600.2.19pSensorDataUsage . . . . .	635
8.600.2.20pSpeedHorizontal . . . . .	635
8.600.2.21pSpeedUnc . . . . .	635
8.600.2.22pSpeedVertical . . . . .	635
8.600.2.23pSvUsedforFix . . . . .	635
8.600.2.24pTechnologyMask . . . . .	635

8.600.2.25	pTimeSrc	635
8.600.2.26	pTimestampUtc	635
8.600.2.27	pTimeUnc	635
8.600.2.28	pVertConfidence	636
8.600.2.29	pVertReliability	636
8.600.2.30	pVertUnc	636
8.600.2.31	sessionId	636
8.600.2.32	sessionStatus	636
8.601	QmiCbkLocSensorStreamingInd Struct Reference	636
8.601.1	Detailed Description	636
8.601.2	Field Documentation	636
8.601.2.1	pAccelAcceptReady	636
8.601.2.2	pAccelTempAcceptReady	636
8.601.2.3	pGyroAcceptReady	636
8.601.2.4	pGyroTempAcceptReady	636
8.602	QmiCbkLocSetExtPowerConfigInd Struct Reference	636
8.602.1	Detailed Description	637
8.602.2	Field Documentation	637
8.602.2.1	status	637
8.603	QmiCbkNasLTECphyCaInfo Struct Reference	637
8.603.1	Detailed Description	637
8.603.2	Field Documentation	638
8.603.2.1	sPhyCaAggPcellInfo	638
8.603.2.2	sPhyCaAggScellIDBw	638
8.603.2.3	sPhyCaAggScellIndex	638
8.603.2.4	sPhyCaAggScellIndType	638
8.603.2.5	sPhyCaAggScellInfo	638
8.604	QmiCbkSwiOmaDmEventStatusReportInd Struct Reference	638
8.604.1	Field Documentation	638
8.604.1.1	SITlv	638
8.605	QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference	638
8.605.1	Field Documentation	638
8.605.1.1	SITlv	638
8.606	QmiCbkTmdMitiLvlRptInd Struct Reference	638
8.606.1	Detailed Description	638
8.606.2	Field Documentation	639
8.606.2.1	currentMitigationLvl	639
8.606.2.2	MitigationDevInfo	639
8.607	QmiCbkWdsStatisticsIndState Struct Reference	639
8.607.1	Detailed Description	639

8.607.2 Field Documentation . . . . .	639
8.607.2.1 RxDropConutTlv . . . . .	640
8.607.2.2 RxOkByteCountTlv . . . . .	640
8.607.2.3 RxOkConutTlv . . . . .	640
8.607.2.4 TxDropConutTlv . . . . .	640
8.607.2.5 TxOkByteCountTlv . . . . .	640
8.607.2.6 TxOkConutTlv . . . . .	640
8.608qmifwinfo_s Struct Reference . . . . .	640
8.608.1 Detailed Description . . . . .	640
8.608.2 Field Documentation . . . . .	640
8.608.2.1 dev . . . . .	641
8.608.2.2 g . . . . .	641
8.608.2.3 s . . . . .	641
8.609QmiNas3GppNetworkInfo Struct Reference . . . . .	641
8.609.1 Detailed Description . . . . .	641
8.609.2 Field Documentation . . . . .	642
8.609.2.1 pDescription . . . . .	642
8.609.2.2 pForbidden . . . . .	642
8.609.2.3 plnUse . . . . .	642
8.609.2.4 pMCC . . . . .	642
8.609.2.5 pMNC . . . . .	642
8.609.2.6 pPreferred . . . . .	642
8.609.2.7 pRoaming . . . . .	642
8.610QmiNasGetRFBandInfoResp Struct Reference . . . . .	642
8.610.1 Field Documentation . . . . .	642
8.610.1.1 plnInstancesSize . . . . .	642
8.610.1.2 pRFBandInfoElements . . . . .	642
8.610.1.3 results . . . . .	642
8.611QmiNasPerformNetworkScanResp Struct Reference . . . . .	642
8.611.1 Field Documentation . . . . .	643
8.611.1.1 plnInstances . . . . .	643
8.611.1.2 plnInstanceSize . . . . .	643
8.611.1.3 results . . . . .	643
8.612qmiSmsMessageList Struct Reference . . . . .	643
8.612.1 Detailed Description . . . . .	643
8.612.2 Field Documentation . . . . .	643
8.612.2.1 messageIndex . . . . .	643
8.612.2.2 messageTag . . . . .	643
8.613qmiWSDSDataBearerTechnology Struct Reference . . . . .	643
8.613.1 Detailed Description . . . . .	643

8.613.2 Field Documentation . . . . .	643
8.613.2.1 currentNetwork . . . . .	643
8.613.2.2 ratMask . . . . .	644
8.613.2.3 soMask . . . . .	644
8.614QmiWdsIpAddressInfo Struct Reference . . . . .	644
8.614.1 Detailed Description . . . . .	644
8.614.2 Field Documentation . . . . .	644
8.614.2.1 pIPAddressV4 . . . . .	644
8.614.2.2 pIPAddressV6 . . . . .	644
8.614.2.3 pIPv6prefixlen . . . . .	644
8.615qmiWdsRunTimeSettings Struct Reference . . . . .	644
8.615.1 Detailed Description . . . . .	645
8.615.2 Field Documentation . . . . .	647
8.615.2.1 pAPNName . . . . .	647
8.615.2.2 pAuthentication . . . . .	647
8.615.2.3 pDomainList . . . . .	647
8.615.2.4 pGPRSGrantedQoS . . . . .	647
8.615.2.5 pGWAddressV4 . . . . .	647
8.615.2.6 pIMCNflag . . . . .	647
8.615.2.7 pIPAddressV4 . . . . .	647
8.615.2.8 pIPFamilyPreference . . . . .	647
8.615.2.9 pIPv6AddrInfo . . . . .	647
8.615.2.10pIPv6GWAddrInfo . . . . .	647
8.615.2.11pMtu . . . . .	647
8.615.2.12pPCSCFAddrPCO . . . . .	647
8.615.2.13pPCSCFFQDNAddrList . . . . .	647
8.615.2.14pPDPTType . . . . .	648
8.615.2.15pPrimaryDNSV4 . . . . .	648
8.615.2.16pPrimaryDNSV6 . . . . .	648
8.615.2.17pProfileID . . . . .	648
8.615.2.18pProfileName . . . . .	648
8.615.2.19pSecondaryDNSV4 . . . . .	648
8.615.2.20pSecondaryDNSV6 . . . . .	648
8.615.2.21pServerAddrList . . . . .	648
8.615.2.22pSubnetMaskV4 . . . . .	648
8.615.2.23pTechnology . . . . .	648
8.615.2.24pUMTSGrantedQoS . . . . .	648
8.615.2.25pUsername . . . . .	648
8.616QosClassID Struct Reference . . . . .	648
8.616.1 Detailed Description . . . . .	648

8.616.2 Field Documentation	649
8.616.2.1 gDIBitRate	649
8.616.2.2 gUIBitRate	649
8.616.2.3 maxDIBitRate	649
8.616.2.4 maxUIBitRate	649
8.616.2.5 QCI	649
8.617QosEventInfo Struct Reference	649
8.617.1 Detailed Description	649
8.617.2 Field Documentation	650
8.617.2.1 pDataBearer	650
8.617.2.2 pPacketsCountRX	650
8.617.2.3 pPacketsCountTX	650
8.617.2.4 pTotalBytesRX	650
8.617.2.5 pTotalBytesTX	651
8.618QosFlowInfo Struct Reference	651
8.618.1 Detailed Description	651
8.618.2 Field Documentation	651
8.618.2.1 pBearerID	651
8.618.2.2 pQFlowState	651
8.618.2.3 pRxQFilter	651
8.618.2.4 pRxQFlowGranted	651
8.618.2.5 pTxQFilter	651
8.618.2.6 pTxQFlowGranted	652
8.619QosFlowInfoState Struct Reference	652
8.619.1 Detailed Description	652
8.619.2 Field Documentation	652
8.619.2.1 id	652
8.619.2.2 isNewFlow	652
8.619.2.3 state	652
8.620QosMap Struct Reference	652
8.620.1 Detailed Description	652
8.620.2 Field Documentation	653
8.620.2.1 dscp	653
8.620.2.2 qos_id	653
8.620.2.3 state	653
8.621RankIndicatorInd Struct Reference	653
8.621.1 Field Documentation	653
8.621.1.1 Count1	653
8.621.1.2 Count2	653
8.622readResult Struct Reference	653

8.622.1 Detailed Description . . . . .	653
8.622.2 Field Documentation . . . . .	654
8.622.2.1 content . . . . .	654
8.622.2.2 contentLen . . . . .	654
8.623readTransparentInfo Struct Reference . . . . .	654
8.623.1 Detailed Description . . . . .	654
8.623.2 Field Documentation . . . . .	654
8.623.2.1 length . . . . .	654
8.623.2.2 offset . . . . .	654
8.624redirNumInfo Struct Reference . . . . .	654
8.624.1 Detailed Description . . . . .	654
8.624.2 Field Documentation . . . . .	656
8.624.2.1 number . . . . .	656
8.624.2.2 numLen . . . . .	656
8.624.2.3 numPlan . . . . .	656
8.624.2.4 numType . . . . .	656
8.624.2.5 PI . . . . .	656
8.624.2.6 reason . . . . .	656
8.624.2.7 SI . . . . .	656
8.625registerRefresh Struct Reference . . . . .	656
8.625.1 Detailed Description . . . . .	656
8.625.2 Field Documentation . . . . .	657
8.625.2.1 arrfileInfo . . . . .	657
8.625.2.2 numFiles . . . . .	657
8.625.2.3 registerFlag . . . . .	657
8.625.2.4 voteForInit . . . . .	657
8.626remainingRetries Struct Reference . . . . .	657
8.626.1 Detailed Description . . . . .	657
8.626.2 Field Documentation . . . . .	657
8.626.2.1 unblockLeft . . . . .	657
8.626.2.2 verifyLeft . . . . .	657
8.627remotePartyName Struct Reference . . . . .	657
8.627.1 Detailed Description . . . . .	658
8.627.2 Field Documentation . . . . .	658
8.627.2.1 callerName . . . . .	658
8.627.2.2 codingScheme . . . . .	658
8.627.2.3 nameLen . . . . .	658
8.627.2.4 namePI . . . . .	658
8.628remotePartyNum Struct Reference . . . . .	658
8.628.1 Detailed Description . . . . .	658

8.628.2 Field Documentation . . . . .	659
8.628.2.1 numLen . . . . .	659
8.628.2.2 presentationInd . . . . .	659
8.628.2.3 remPartyNumber . . . . .	659
8.629ReqFieldsList Struct Reference . . . . .	659
8.629.1 Detailed Description . . . . .	659
8.629.2 Field Documentation . . . . .	659
8.629.2.1 requestFields . . . . .	659
8.629.2.2 requestFieldsLen . . . . .	660
8.630RespFieldsList Struct Reference . . . . .	660
8.630.1 Detailed Description . . . . .	660
8.630.2 Field Documentation . . . . .	660
8.630.2.1 responseFields . . . . .	660
8.630.2.2 responseFieldsLen . . . . .	660
8.631RFBandInfoElements Struct Reference . . . . .	660
8.631.1 Detailed Description . . . . .	660
8.631.2 Field Documentation . . . . .	661
8.631.2.1 activeBandClass . . . . .	661
8.631.2.2 activeBandClass . . . . .	661
8.631.2.3 activeChannel . . . . .	661
8.631.2.4 activeChannel . . . . .	661
8.631.2.5 radiolInterface . . . . .	661
8.631.2.6 radiolInterface . . . . .	661
8.632rmTrasnrStaticsReq Struct Reference . . . . .	661
8.632.1 Detailed Description . . . . .	661
8.632.2 Field Documentation . . . . .	661
8.632.2.1 bResetStatistics . . . . .	661
8.632.2.2 ulMask . . . . .	661
8.633roamIndList Struct Reference . . . . .	661
8.633.1 Detailed Description . . . . .	662
8.633.2 Field Documentation . . . . .	662
8.633.2.1 numInstances . . . . .	662
8.633.2.2 radiolInterface . . . . .	662
8.633.2.3 roamIndicator . . . . .	662
8.634RoamingInfo Struct Reference . . . . .	662
8.634.1 Field Documentation . . . . .	662
8.634.1.1 roaming_ind . . . . .	662
8.634.1.2 TlvPresent . . . . .	662
8.635roamTimer Struct Reference . . . . .	662
8.635.1 Detailed Description . . . . .	663

8.635.2 Field Documentation . . . . .	663
8.635.2.1 namID . . . . .	663
8.635.2.2 roamTimerValue . . . . .	663
8.636RSRPThresh Struct Reference . . . . .	663
8.636.1 Detailed Description . . . . .	663
8.636.2 Field Documentation . . . . .	664
8.636.2.1 pRSRPThresList . . . . .	664
8.636.2.2 RSRPThresListLen . . . . .	664
8.637rsrqInformation Struct Reference . . . . .	664
8.637.1 Detailed Description . . . . .	664
8.637.2 Field Documentation . . . . .	664
8.637.2.1 radiolf . . . . .	664
8.637.2.2 rsrq . . . . .	664
8.638RSRQThresh Struct Reference . . . . .	664
8.638.1 Detailed Description . . . . .	665
8.638.2 Field Documentation . . . . .	665
8.638.2.1 pRSRQThresList . . . . .	665
8.638.2.2 RSRQThresListLen . . . . .	665
8.639RSSIThresh Struct Reference . . . . .	665
8.639.1 Detailed Description . . . . .	665
8.639.2 Field Documentation . . . . .	666
8.639.2.1 pRSSIThresList . . . . .	666
8.639.2.2 RSSIThresListLen . . . . .	666
8.640RXAGCList Struct Reference . . . . .	666
8.640.1 Detailed Description . . . . .	666
8.640.2 Field Documentation . . . . .	667
8.640.2.1 pRXAIG . . . . .	667
8.640.2.2 pRXComprSlope . . . . .	667
8.640.2.3 pRXComprThres . . . . .	667
8.640.2.4 pRXExpSlope . . . . .	667
8.640.2.5 pRXExpThres . . . . .	667
8.640.2.6 pRXStaticGain . . . . .	667
8.641RXAVCList Struct Reference . . . . .	667
8.641.1 Detailed Description . . . . .	667
8.641.2 Field Documentation . . . . .	667
8.641.2.1 pAVRXAVCHearoom . . . . .	667
8.641.2.2 pAVRXAVCSens . . . . .	667
8.642rxInfo Struct Reference . . . . .	667
8.642.1 Detailed Description . . . . .	668
8.642.2 Field Documentation . . . . .	668

8.642.2.1 ecio . . . . .	668
8.642.2.2 isRadioTuned . . . . .	668
8.642.2.3 phase . . . . .	668
8.642.2.4 rscp . . . . .	668
8.642.2.5 rsrp . . . . .	668
8.642.2.6 rxPower . . . . .	668
8.643RXPCMIIRFiltr Struct Reference . . . . .	669
8.643.1 Detailed Description . . . . .	669
8.643.2 Field Documentation . . . . .	670
8.643.2.1 pFlag . . . . .	670
8.643.2.2 pStage0Val . . . . .	670
8.643.2.3 pStage1Val . . . . .	670
8.643.2.4 pStage2Val . . . . .	670
8.643.2.5 pStage3Val . . . . .	670
8.643.2.6 pStage4Val . . . . .	670
8.643.2.7 pStageCnt . . . . .	670
8.644rxSignalStrengthListElement Struct Reference . . . . .	670
8.644.1 Detailed Description . . . . .	670
8.644.2 Field Documentation . . . . .	671
8.644.2.1 radiolf . . . . .	671
8.644.2.2 rxSignalStrength . . . . .	671
8.645sApnExtraParams Struct Reference . . . . .	671
8.645.1 Detailed Description . . . . .	671
8.645.2 Field Documentation . . . . .	672
8.645.2.1 ambr_dl . . . . .	672
8.645.2.2 ambr_dl_ext . . . . .	672
8.645.2.3 ambr_dl_ext2 . . . . .	672
8.645.2.4 ambr_ul . . . . .	672
8.645.2.5 ambr_ul_ext . . . . .	672
8.645.2.6 ambr_ul_ext2 . . . . .	672
8.645.2.7 apnId . . . . .	672
8.646satelliteInfo Struct Reference . . . . .	672
8.646.1 Detailed Description . . . . .	672
8.646.2 Field Documentation . . . . .	674
8.646.2.1 azimuth . . . . .	674
8.646.2.2 elevation . . . . .	674
8.646.2.3 gnssSvId . . . . .	674
8.646.2.4 healthStatus . . . . .	674
8.646.2.5 snr . . . . .	674
8.646.2.6 svInfoMask . . . . .	674

8.646.2.7 svListLen . . . . .	674
8.646.2.8 svStatus . . . . .	674
8.646.2.9 system . . . . .	674
8.646.2.10validMask . . . . .	674
8.647sensorData Struct Reference . . . . .	674
8.647.1 Detailed Description . . . . .	675
8.647.2 Field Documentation . . . . .	675
8.647.2.1 flags . . . . .	676
8.647.2.2 sensorDataLen . . . . .	676
8.647.2.3 timeOfFirstSample . . . . .	676
8.647.2.4 timeOffset . . . . .	676
8.647.2.5 xAxis . . . . .	676
8.647.2.6 yAxis . . . . .	676
8.647.2.7 zAxis . . . . .	676
8.648sensorDataUsage_s Struct Reference . . . . .	676
8.648.1 Detailed Description . . . . .	676
8.648.2 Field Documentation . . . . .	676
8.648.2.1 aidingIndicatorMask . . . . .	676
8.648.2.2 usageMask . . . . .	676
8.649serialNumbersInfo Struct Reference . . . . .	677
8.649.1 Detailed Description . . . . .	677
8.649.2 Field Documentation . . . . .	677
8.649.2.1 esnSize . . . . .	677
8.649.2.2 imeiSize . . . . .	678
8.649.2.3 imeiSvnSize . . . . .	678
8.649.2.4 meidSize . . . . .	678
8.649.2.5 pESNString . . . . .	678
8.649.2.6 plMEIString . . . . .	678
8.649.2.7 plmeiSvnString . . . . .	678
8.649.2.8 pMEIDString . . . . .	678
8.650serviceProviderName Struct Reference . . . . .	678
8.650.1 Detailed Description . . . . .	678
8.650.2 Field Documentation . . . . .	678
8.650.2.1 displayCondition . . . . .	678
8.650.2.2 spn . . . . .	678
8.650.2.3 spnLength . . . . .	678
8.651ServingSystemInfo Struct Reference . . . . .	678
8.651.1 Detailed Description . . . . .	679
8.651.2 Field Documentation . . . . .	680
8.651.2.1 csAttachState . . . . .	680

8.651.2.2	hdrPersonality	680
8.651.2.3	psAttachState	680
8.651.2.4	radiolInterfaceList	680
8.651.2.5	radiolInterfaceNo	680
8.651.2.6	registrationState	680
8.651.2.7	selectedNetwork	680
8.652	servSystem Struct Reference	680
8.652.1	Detailed Description	680
8.652.2	Field Documentation	681
8.652.2.1	csAttachState	681
8.652.2.2	numRadiolInterfaces	681
8.652.2.3	psAttachState	681
8.652.2.4	radiolInterface	681
8.652.2.5	regState	681
8.652.2.6	selNetwork	681
8.653	sessionInfo Union Reference	681
8.653.1	Detailed Description	682
8.653.2	Field Documentation	682
8.653.2.1	omaDmConfig	682
8.653.2.2	omaDmFota	682
8.653.2.3	omaDmNotifications	682
8.654	sessionInfoExt Union Reference	682
8.654.1	Detailed Description	682
8.654.2	Field Documentation	682
8.654.2.1	omaDmConfig	682
8.654.2.2	omaDmFota	682
8.655	sessionInfoTlv Struct Reference	682
8.655.1	Detailed Description	682
8.655.2	Field Documentation	683
8.655.2.1	sessionInfo	683
8.655.2.2	sessionType	683
8.655.2.3	TlvPresent	683
8.656	sessionInfoTlvExt Struct Reference	683
8.656.1	Detailed Description	683
8.656.2	Field Documentation	683
8.656.2.1	sessionInfo	683
8.656.2.2	sessionType	683
8.656.2.3	TlvPresent	683
8.657	SetAudioPathConfigReq Struct Reference	683
8.657.1	Detailed Description	684

8.657.2 Field Documentation	685
8.657.2.1 pCodecSTGain	685
8.657.2.2 pDTMFTXGain	685
8.657.2.3 pECMode	685
8.657.2.4 pNSEnable	685
8.657.2.5 Profile	685
8.657.2.6 pRXAGCList	685
8.657.2.7 pRXAVCAGCSwitch	685
8.657.2.8 pRXAVCList	685
8.657.2.9 pRXPCMIIRFiltr	685
8.657.2.10pTXAGCList	685
8.657.2.11pTXAVCSwitch	685
8.657.2.12pTXGain	685
8.657.2.13pTXPCMIIRFiltr	685
8.658SetAudioProfileReq Struct Reference	685
8.658.1 Detailed Description	685
8.658.2 Field Documentation	686
8.658.2.1 EarMute	686
8.658.2.2 Generator	686
8.658.2.3 MicMute	686
8.658.2.4 Profile	686
8.658.2.5 Volume	686
8.659SetAudioVolTLBConfigReq Struct Reference	686
8.659.1 Detailed Description	687
8.659.2 Field Documentation	687
8.659.2.1 Generator	687
8.659.2.2 Item	687
8.659.2.3 Profile	687
8.659.2.4 Volume	687
8.659.2.5 VolValue	687
8.660SetAudioVolTLBConfigResp Struct Reference	687
8.660.1 Detailed Description	688
8.660.2 Field Documentation	688
8.660.2.1 ResCode	688
8.661setCustomSettingV2 Struct Reference	688
8.661.1 Detailed Description	688
8.661.2 Field Documentation	688
8.661.2.1 cust_id	688
8.661.2.2 cust_value	688
8.661.2.3 value_length	688

8.662setDyingGaspCfg Struct Reference . . . . .	688
8.662.1 Detailed Description . . . . .	689
8.662.2 Field Documentation . . . . .	689
8.662.2.1 pDestSMSContent . . . . .	689
8.662.2.2 pDestSMSNum . . . . .	689
8.663SetIMSSMSConfigReq Struct Reference . . . . .	689
8.663.1 Detailed Description . . . . .	689
8.663.2 Field Documentation . . . . .	690
8.663.2.1 pPhoneCtxtURI . . . . .	690
8.663.2.2 pPhoneCtxtURILen . . . . .	690
8.663.2.3 pSMSFormat . . . . .	690
8.663.2.4 pSMSOverIPNwInd . . . . .	690
8.664SetIMSSMSConfigResp Struct Reference . . . . .	690
8.664.1 Detailed Description . . . . .	690
8.664.2 Field Documentation . . . . .	690
8.664.2.1 pSettingResp . . . . .	690
8.665SetIMSUserConfigReq Struct Reference . . . . .	690
8.665.1 Detailed Description . . . . .	690
8.665.2 Field Documentation . . . . .	690
8.665.2.1 pIMSDomain . . . . .	691
8.665.2.2 pIMSDomainLen . . . . .	691
8.666SetIMSUserConfigResp Struct Reference . . . . .	691
8.666.1 Detailed Description . . . . .	691
8.666.2 Field Documentation . . . . .	691
8.666.2.1 pSettingResp . . . . .	691
8.667SetIMSVoIPConfigReq Struct Reference . . . . .	691
8.667.1 Detailed Description . . . . .	691
8.667.2 Field Documentation . . . . .	693
8.667.2.1 pAmrMode . . . . .	693
8.667.2.2 pAmrOctetAligned . . . . .	693
8.667.2.3 pAmrWbEnable . . . . .	693
8.667.2.4 pAmrWBMode . . . . .	693
8.667.2.5 pAmrWBOctetAligned . . . . .	693
8.667.2.6 pMinSessionExpiryTimer . . . . .	693
8.667.2.7 pRingBackTimer . . . . .	693
8.667.2.8 pRingingTimer . . . . .	693
8.667.2.9 pRTPRTCPInactTimer . . . . .	693
8.667.2.10pScrAmrEnable . . . . .	693
8.667.2.11pScrAmrWbEnable . . . . .	693
8.667.2.12pSessionExpiryTimer . . . . .	693

8.668SetIMSVoIPConfigResp Struct Reference	693
8.668.1 Detailed Description	694
8.668.2 Field Documentation	694
8.668.2.1 pSettingResp	694
8.669SetM2MAudioAVCFGReq Struct Reference	694
8.669.1 Detailed Description	694
8.669.2 Field Documentation	694
8.669.2.1 Device	694
8.669.2.2 PIFACEId	694
8.669.2.3 pPCMPParams	695
8.669.2.4 Profile	695
8.670SetM2MAudioLPBKReq Struct Reference	695
8.670.1 Detailed Description	695
8.670.2 Field Documentation	695
8.670.2.1 Enable	695
8.671SetM2MAudioProfileReq Struct Reference	695
8.671.1 Detailed Description	695
8.671.2 Field Documentation	696
8.671.2.1 pCwtMute	696
8.671.2.2 pEarMute	696
8.671.2.3 pGenerator	696
8.671.2.4 pMicMute	696
8.671.2.5 Profile	696
8.671.2.6 pVolume	696
8.672SetM2MAudioVolumeReq Struct Reference	696
8.672.1 Detailed Description	696
8.672.2 Field Documentation	697
8.672.2.1 Generator	697
8.672.2.2 Level	697
8.672.2.3 Profile	697
8.673SetM2MAVMuteReq Struct Reference	697
8.673.1 Detailed Description	697
8.673.2 Field Documentation	697
8.673.2.1 EarMute	698
8.673.2.2 MicMute	698
8.673.2.3 pCwtMute	698
8.673.2.4 Profile	698
8.674SetM2MSpkrGainReq Struct Reference	698
8.674.1 Detailed Description	698
8.674.2 Field Documentation	698

8.674.2.1 Profile . . . . .	698
8.674.2.2 Value . . . . .	698
8.675setPINProtection Struct Reference . . . . .	698
8.675.1 Detailed Description . . . . .	698
8.675.2 Field Documentation . . . . .	699
8.675.2.1 pinID . . . . .	699
8.675.2.2 pinLength . . . . .	699
8.675.2.3 pinOperation . . . . .	699
8.675.2.4 pinValue . . . . .	699
8.676SetRegMgrConfigReq Struct Reference . . . . .	699
8.676.1 Detailed Description . . . . .	699
8.676.2 Field Documentation . . . . .	700
8.676.2.1 pCSCFPortName . . . . .	700
8.676.2.2 pCSCFPortNameLen . . . . .	700
8.676.2.3 pIMSTestMode . . . . .	700
8.676.2.4 pPriCSCFPort . . . . .	700
8.677SetRegMgrConfigResp Struct Reference . . . . .	700
8.677.1 Detailed Description . . . . .	700
8.677.2 Field Documentation . . . . .	700
8.677.2.1 pSettingResp . . . . .	700
8.678setSignalStrengthInfo Struct Reference . . . . .	700
8.678.1 Detailed Description . . . . .	701
8.678.2 Field Documentation . . . . .	704
8.678.2.1 pCDMAECIODelta . . . . .	704
8.678.2.2 pCDMAECIOThresh . . . . .	704
8.678.2.3 pCDMARSSIDelta . . . . .	704
8.678.2.4 pCDMARSSIThresh . . . . .	704
8.678.2.5 pGSMRSSIDelta . . . . .	704
8.678.2.6 pGSMRSSIThresh . . . . .	704
8.678.2.7 pHRECIODelta . . . . .	704
8.678.2.8 pHRECIOThresh . . . . .	704
8.678.2.9 pHDRIODelta . . . . .	704
8.678.2.10pHDRIOThresh . . . . .	704
8.678.2.11pHDRRSSIDelta . . . . .	704
8.678.2.12pHDRRSSIThresh . . . . .	704
8.678.2.13pHDRSINRDelta . . . . .	704
8.678.2.14pHDRSINRThresh . . . . .	704
8.678.2.15pLTERSRPDelta . . . . .	704
8.678.2.16pLTERSRPThresh . . . . .	704
8.678.2.17pLTERSQRDelta . . . . .	704

8.678.2.18pLTERSRQThresh . . . . .	705
8.678.2.19pLTERSSIDelta . . . . .	705
8.678.2.20pLTERSSIThresh . . . . .	705
8.678.2.21pLTESigRptConfig . . . . .	705
8.678.2.22pLTESNRDelta . . . . .	705
8.678.2.23pLTESNRThresh . . . . .	705
8.678.2.24pTDSCDMAECIODelta . . . . .	705
8.678.2.25pTDSCDMAECIOThresh . . . . .	705
8.678.2.26pTDSCDMARSCPDelta . . . . .	705
8.678.2.27pTDSCDMARSCPThresh . . . . .	705
8.678.2.28pTDSCDMARSSIDelta . . . . .	705
8.678.2.29pTDSCDMARSSIThresh . . . . .	705
8.678.2.30pTDSCDMASINRDelta . . . . .	705
8.678.2.31pTDSCDMASINRThresh . . . . .	705
8.678.2.32pWCDMAECIODelta . . . . .	705
8.678.2.33pWCDMAECIOThresh . . . . .	705
8.678.2.34pWCDMARSSIDelta . . . . .	705
8.678.2.35pWCDMARSSIThresh . . . . .	705
8.679SetSIPConfigReq Struct Reference . . . . .	705
8.679.1 Detailed Description . . . . .	705
8.679.2 Field Documentation . . . . .	706
8.679.2.1 pSigCompEnabled . . . . .	706
8.679.2.2 pSIPLocalPort . . . . .	706
8.679.2.3 pSubscribeTimer . . . . .	706
8.679.2.4 pTimerSIPReg . . . . .	706
8.679.2.5 pTimerT1 . . . . .	706
8.679.2.6 pTimerT2 . . . . .	706
8.679.2.7 pTimerTf . . . . .	706
8.680SetSIPConfigResp Struct Reference . . . . .	706
8.680.1 Detailed Description . . . . .	706
8.680.2 Field Documentation . . . . .	707
8.680.2.1 pSettingResp . . . . .	707
8.681sGetDeviceSeriesResult Struct Reference . . . . .	707
8.681.1 Detailed Description . . . . .	707
8.681.2 Field Documentation . . . . .	707
8.681.2.1 eDevice . . . . .	707
8.681.2.2 uResult . . . . .	707
8.682sidNid Struct Reference . . . . .	707
8.682.1 Detailed Description . . . . .	707
8.682.2 Field Documentation . . . . .	708

8.682.2.1 nid . . . . .	708
8.682.2.2 sid . . . . .	708
8.683sigInfo Struct Reference . . . . .	708
8.683.1 Detailed Description . . . . .	708
8.683.2 Field Documentation . . . . .	709
8.683.2.1 pECIOThresh . . . . .	709
8.683.2.2 pHDRSINRThresh . . . . .	709
8.683.2.3 pIOThresh . . . . .	709
8.683.2.4 pLTESigRptCfg . . . . .	709
8.683.2.5 pLTESNRThresh . . . . .	709
8.683.2.6 pRSRPThresh . . . . .	709
8.683.2.7 pRSRQThresh . . . . .	709
8.683.2.8 pRSSIThresh . . . . .	709
8.683.2.9 pTDSCDMASINRCONFThresh . . . . .	709
8.684signalInfo Struct Reference . . . . .	709
8.684.1 Detailed Description . . . . .	709
8.684.2 Field Documentation . . . . .	710
8.684.2.1 alertPitch . . . . .	710
8.684.2.2 signal . . . . .	710
8.684.2.3 signalType . . . . .	710
8.685SignalStrengthDataType Struct Reference . . . . .	710
8.685.1 Field Documentation . . . . .	710
8.685.1.1 thresholds . . . . .	710
8.685.1.2 thresholdsSize . . . . .	710
8.686slot_t Struct Reference . . . . .	710
8.686.1 Detailed Description . . . . .	710
8.686.2 Field Documentation . . . . .	711
8.686.2.1 bICCID . . . . .	711
8.686.2.2 bICCIDLength . . . . .	711
8.686.2.3 bLogicalSlot . . . . .	711
8.686.2.4 uPhyCardStatus . . . . .	711
8.686.2.5 uPhySlotStatus . . . . .	711
8.687slotInf Struct Reference . . . . .	711
8.687.1 Detailed Description . . . . .	711
8.687.2 Field Documentation . . . . .	712
8.687.2.1 AppStatus . . . . .	712
8.687.2.2 cardState . . . . .	712
8.687.2.3 errorState . . . . .	712
8.687.2.4 numApp . . . . .	712
8.687.2.5 upinRetries . . . . .	713

8.687.2.6 upinState . . . . .	713
8.687.2.7 upukRetries . . . . .	713
8.688slotInfo Struct Reference . . . . .	713
8.688.1 Detailed Description . . . . .	713
8.688.2 Field Documentation . . . . .	714
8.688.2.1 AppStatus . . . . .	714
8.688.2.2 cardState . . . . .	714
8.688.2.3 errorState . . . . .	714
8.688.2.4 numApp . . . . .	714
8.688.2.5 upinRetries . . . . .	714
8.688.2.6 upinState . . . . .	714
8.688.2.7 upukRetries . . . . .	714
8.689slots_t Struct Reference . . . . .	714
8.689.1 Field Documentation . . . . .	714
8.689.1.1 uimSlotStatus . . . . .	714
8.690slqsautoconnect Struct Reference . . . . .	714
8.690.1 Detailed Description . . . . .	715
8.690.2 Field Documentation . . . . .	715
8.690.2.1 acroamsetting . . . . .	715
8.690.2.2 acsetting . . . . .	715
8.690.2.3 action . . . . .	715
8.691SLQSDeleteProfileParams Struct Reference . . . . .	715
8.691.1 Detailed Description . . . . .	715
8.691.2 Field Documentation . . . . .	716
8.691.2.1 profileIndex . . . . .	716
8.691.2.2 profileType . . . . .	716
8.692slqsfwinfo_s Struct Reference . . . . .	716
8.692.1 Detailed Description . . . . .	716
8.692.2 Field Documentation . . . . .	717
8.692.2.1 appversion_str . . . . .	717
8.692.2.2 bootversion_str . . . . .	717
8.692.2.3 carrier_str . . . . .	717
8.692.2.4 cur_carr_name . . . . .	717
8.692.2.5 cur_carr_rev . . . . .	717
8.692.2.6 modelid_str . . . . .	717
8.692.2.7 packageid_str . . . . .	717
8.692.2.8 priversion_str . . . . .	717
8.692.2.9 sku_str . . . . .	717
8.693SlqsNas3GppNetworkInfo Struct Reference . . . . .	717
8.693.1 Detailed Description . . . . .	717

8.693.2 Field Documentation . . . . .	718
8.693.2.1 Description . . . . .	718
8.693.2.2 Forbidden . . . . .	718
8.693.2.3 InUse . . . . .	718
8.693.2.4 MCC . . . . .	718
8.693.2.5 MNC . . . . .	718
8.693.2.6 Preferred . . . . .	718
8.693.2.7 Roaming . . . . .	718
8.694SlqsNasPcsDigit Struct Reference . . . . .	718
8.694.1 Detailed Description . . . . .	719
8.694.2 Field Documentation . . . . .	719
8.694.2.1 includes_pcs_digit . . . . .	719
8.694.2.2 MCC . . . . .	719
8.694.2.3 MNC . . . . .	719
8.695slqssendasynsmsparams_s Struct Reference . . . . .	719
8.695.1 Detailed Description . . . . .	719
8.695.2 Field Documentation . . . . .	721
8.695.2.1 messageFormat . . . . .	721
8.695.2.2 messageSize . . . . .	721
8.695.2.3 pFollowOnDC . . . . .	721
8.695.2.4 pForceOnDC . . . . .	721
8.695.2.5 pLinktimer . . . . .	721
8.695.2.6 pMessage . . . . .	721
8.695.2.7 pRetryMessage . . . . .	721
8.695.2.8 pRetryMessageId . . . . .	721
8.695.2.9 pServiceOption . . . . .	721
8.695.2.10pSmsOnIms . . . . .	721
8.695.2.11pUserData . . . . .	721
8.696slqssendsmsparams_s Struct Reference . . . . .	721
8.696.1 Detailed Description . . . . .	721
8.696.2 Field Documentation . . . . .	722
8.696.2.1 messageFailureCode . . . . .	722
8.696.2.2 messageFormat . . . . .	722
8.696.2.3 messageId . . . . .	722
8.696.2.4 messageSize . . . . .	722
8.696.2.5 pLinktimer . . . . .	722
8.696.2.6 pMessage . . . . .	722
8.696.2.7 pSmsOnIms . . . . .	722
8.697slqsSessionStateInfo Struct Reference . . . . .	722
8.697.1 Detailed Description . . . . .	723

8.697.2 Field Documentation	723
8.697.2.1 pQmiInterfaceInfo	723
8.697.2.2 reconfiguration_required	723
8.697.2.3 sessionEndReason	723
8.697.2.4 state	723
8.698slqsSignalStrengthInfo Struct Reference	723
8.698.1 Detailed Description	724
8.698.2 Field Documentation	725
8.698.2.1 ecioList	725
8.698.2.2 ecioListLen	725
8.698.2.3 errorRateList	725
8.698.2.4 errorRateListLen	725
8.698.2.5 lo	725
8.698.2.6 ltersrp	725
8.698.2.7 ltesnr	725
8.698.2.8 rsrqInfo	725
8.698.2.9 rxSignalStrengthList	725
8.698.2.10rxSignalStrengthListLen	725
8.698.2.11signalStrengthReqMask	726
8.698.2.12sinr	726
8.699SLQSSignalStrengthsIndReq Struct Reference	726
8.699.1 Detailed Description	726
8.699.2 Field Documentation	727
8.699.2.1 ecioDelta	727
8.699.2.2 ecioThresholdList	727
8.699.2.3 ecioThresholdListLen	727
8.699.2.4 ioDelta	727
8.699.2.5 lteRsrpDelta	727
8.699.2.6 lteSnrDelta	727
8.699.2.7 rsrqDelta	727
8.699.2.8 rxSignalStrengthDelta	727
8.699.2.9 sinrDelta	727
8.699.2.10sinrThresholdList	727
8.699.2.11sinrThresholdListLen	727
8.700SLQSSignalStrengthsInformation Struct Reference	728
8.700.1 Detailed Description	728
8.700.2 Field Documentation	728
8.700.2.1 ecioInfo	729
8.700.2.2 errorRateInfo	729
8.700.2.3 io	729

8.700.2.4 lteRsrpinfo . . . . .	729
8.700.2.5 lteSnrinfo . . . . .	729
8.700.2.6 rsrqInfo . . . . .	729
8.700.2.7 rxSignalStrengthInfo . . . . .	729
8.700.2.8 sinr . . . . .	729
8.701slqsWdsEventInfo Struct Reference . . . . .	729
8.701.1 Detailed Description . . . . .	729
8.701.2 Field Documentation . . . . .	730
8.701.2.1 pDataBearer . . . . .	730
8.701.2.2 pDormancyStatus . . . . .	730
8.701.2.3 pPacketsCountRX . . . . .	730
8.701.2.4 pPacketsCountTX . . . . .	730
8.701.2.5 pQmiInterfaceInfo . . . . .	731
8.701.2.6 pTotalBytesRX . . . . .	731
8.701.2.7 pTotalBytesTX . . . . .	731
8.702SMSAsyncRawSend_s Struct Reference . . . . .	731
8.702.1 Detailed Description . . . . .	731
8.702.2 Field Documentation . . . . .	732
8.702.2.1 alphaIDLen . . . . .	732
8.702.2.2 causeCode . . . . .	732
8.702.2.3 errorClass . . . . .	732
8.702.2.4 messageId . . . . .	732
8.702.2.5 msgDelFailureCause . . . . .	732
8.702.2.6 msgDelFailureType . . . . .	732
8.702.2.7 pAlphaID . . . . .	732
8.702.2.8 RPCause . . . . .	732
8.702.2.9 sendStatus . . . . .	732
8.702.2.10TPCause . . . . .	732
8.702.2.11userData . . . . .	732
8.703sMSCAddress Struct Reference . . . . .	732
8.703.1 Detailed Description . . . . .	733
8.703.2 Field Documentation . . . . .	733
8.703.2.1 data . . . . .	733
8.703.2.2 length . . . . .	733
8.704SMSCAddress Struct Reference . . . . .	733
8.704.1 Detailed Description . . . . .	733
8.704.2 Field Documentation . . . . .	733
8.704.2.1 data . . . . .	733
8.704.2.2 length . . . . .	733
8.705sMSCAddressTlv Struct Reference . . . . .	733

8.705.1 Detailed Description . . . . .	733
8.705.2 Field Documentation . . . . .	734
8.705.2.1 SMSCInfo . . . . .	734
8.705.2.2 TlvPresent . . . . .	734
8.706sMSEtwsMessage Struct Reference . . . . .	734
8.706.1 Detailed Description . . . . .	734
8.706.2 Field Documentation . . . . .	734
8.706.2.1 data . . . . .	734
8.706.2.2 length . . . . .	734
8.706.2.3 notificationType . . . . .	734
8.707SMSEtwsMessage Struct Reference . . . . .	734
8.707.1 Detailed Description . . . . .	735
8.707.2 Field Documentation . . . . .	735
8.707.2.1 data . . . . .	735
8.707.2.2 length . . . . .	735
8.707.2.3 notificationType . . . . .	735
8.708sMSEtwsMessageTlv Struct Reference . . . . .	735
8.708.1 Detailed Description . . . . .	735
8.708.2 Field Documentation . . . . .	735
8.708.2.1 EtwsMessageInfo . . . . .	735
8.708.2.2 TlvPresent . . . . .	735
8.709sMSEtwsPlmn Struct Reference . . . . .	735
8.709.1 Detailed Description . . . . .	736
8.709.2 Field Documentation . . . . .	736
8.709.2.1 mobileCountryCode . . . . .	736
8.709.2.2 mobileNetworkCode . . . . .	736
8.710SMSEtwsPlmn Struct Reference . . . . .	736
8.710.1 Detailed Description . . . . .	736
8.710.2 Field Documentation . . . . .	736
8.710.2.1 mobileCountryCode . . . . .	736
8.710.2.2 mobileNetworkCode . . . . .	736
8.711SMSEventInfo_s Struct Reference . . . . .	736
8.711.1 Detailed Description . . . . .	737
8.711.2 Field Documentation . . . . .	737
8.711.2.1 pEtwsMessageInfo . . . . .	737
8.711.2.2 pEtwsPlmnInfo . . . . .	737
8.711.2.3 pMessageModelInfo . . . . .	738
8.711.2.4 pMTMessageInfo . . . . .	738
8.711.2.5 pSMSCAddressInfo . . . . .	738
8.711.2.6 pSMSONIMSInfo . . . . .	738

8.711.2.7 pTransferRouteMTMessageInfo . . . . .	738
8.711.2.8 smsEventType . . . . .	738
8.712smsMaxStorageSizeReq Struct Reference . . . . .	738
8.712.1 Detailed Description . . . . .	738
8.712.2 Field Documentation . . . . .	738
8.712.2.1 pMessageMode . . . . .	738
8.712.2.2 storageType . . . . .	738
8.713smsMaxStorageSizeResp Struct Reference . . . . .	738
8.713.1 Detailed Description . . . . .	739
8.713.2 Field Documentation . . . . .	739
8.713.2.1 freeSlots . . . . .	739
8.713.2.2 maxStorageSize . . . . .	739
8.714SMSMemoryInfo Struct Reference . . . . .	739
8.714.1 Detailed Description . . . . .	739
8.714.2 Field Documentation . . . . .	739
8.714.2.1 messageMode . . . . .	739
8.714.2.2 storageType . . . . .	739
8.715sMSMessageMode Struct Reference . . . . .	739
8.715.1 Detailed Description . . . . .	740
8.715.2 Field Documentation . . . . .	740
8.715.2.1 messageMode . . . . .	740
8.716SMSMessageMode Struct Reference . . . . .	740
8.716.1 Detailed Description . . . . .	740
8.716.2 Field Documentation . . . . .	740
8.716.2.1 messageMode . . . . .	740
8.717smsMsgprotocolResp Struct Reference . . . . .	740
8.717.1 Detailed Description . . . . .	740
8.717.2 Field Documentation . . . . .	741
8.717.2.1 msgProtocol . . . . .	741
8.718sMSMTMessage Struct Reference . . . . .	741
8.718.1 Detailed Description . . . . .	741
8.718.2 Field Documentation . . . . .	741
8.718.2.1 messageIndex . . . . .	741
8.718.2.2 storageType . . . . .	741
8.719SMSMTMessage Struct Reference . . . . .	741
8.719.1 Detailed Description . . . . .	741
8.719.2 Field Documentation . . . . .	741
8.719.2.1 messageIndex . . . . .	741
8.719.2.2 storageType . . . . .	741
8.720sMSOnIMS Struct Reference . . . . .	742

8.720.1 Detailed Description	742
8.720.2 Field Documentation	742
8.720.2.1 smsOnIMS	742
8.721 SMSOnIMS Struct Reference	742
8.721.1 Detailed Description	742
8.721.2 Field Documentation	742
8.721.2.1 smsOnIMS	742
8.722 sMSOnIMSTlv Struct Reference	742
8.722.1 Detailed Description	742
8.722.2 Field Documentation	743
8.722.2.1 IMSInfo	743
8.722.2.2 TlvPresent	743
8.723 smsRouteEntry Struct Reference	743
8.723.1 Detailed Description	743
8.723.2 Field Documentation	744
8.723.2.1 messageClass	744
8.723.2.2 messageType	744
8.723.2.3 receiptAction	744
8.723.2.4 routeStorage	744
8.724 smsSetRoutesReq Struct Reference	744
8.724.1 Detailed Description	744
8.724.2 Field Documentation	744
8.724.2.1 numOfRoutes	744
8.724.2.2 pTransferStatusReport	744
8.724.2.3 routeList	744
8.725 sMSTransferRouteMTMessage Struct Reference	745
8.725.1 Detailed Description	745
8.725.2 Field Documentation	745
8.725.2.1 ackIndicator	745
8.725.2.2 data	745
8.725.2.3 format	745
8.725.2.4 length	745
8.725.2.5 transactionID	745
8.726 SMSTransferRouteMTMessage Struct Reference	745
8.726.1 Detailed Description	746
8.726.2 Field Documentation	746
8.726.2.1 ackIndicator	746
8.726.2.2 data	746
8.726.2.3 format	746
8.726.2.4 length	746

8.726.2.5 transactionID . . . . .	746
8.727sQosFlowStat Struct Reference . . . . .	746
8.727.1 Detailed Description . . . . .	746
8.727.2 Field Documentation . . . . .	747
8.727.2.1 bearerId . . . . .	747
8.727.2.2 tx_bytes . . . . .	747
8.727.2.3 tx_bytes_drp . . . . .	747
8.727.2.4 tx_pkt . . . . .	747
8.727.2.5 tx_pkt_drp . . . . .	747
8.728sQosStat Struct Reference . . . . .	747
8.728.1 Detailed Description . . . . .	747
8.728.2 Field Documentation . . . . .	748
8.728.2.1 apnId . . . . .	748
8.728.2.2 numQosFlow . . . . .	748
8.728.2.3 qosFlow . . . . .	748
8.728.2.4 total_rx_bytes . . . . .	748
8.728.2.5 total_rx_pkt . . . . .	748
8.728.2.6 total_tx_bytes . . . . .	748
8.728.2.7 total_tx_bytes_drp . . . . .	748
8.728.2.8 total_tx_pkt . . . . .	748
8.728.2.9 total_tx_pkt_drp . . . . .	748
8.729SrvStatusInfo Struct Reference . . . . .	748
8.729.1 Detailed Description . . . . .	749
8.729.2 Field Documentation . . . . .	749
8.729.2.1 isPrefDataPath . . . . .	749
8.729.2.2 srvStatus . . . . .	749
8.730ssdatasession_params Struct Reference . . . . .	749
8.730.1 Detailed Description . . . . .	750
8.730.2 Field Documentation . . . . .	751
8.730.2.1 action . . . . .	751
8.730.2.2 failureReason . . . . .	751
8.730.2.3 failureReasonv4 . . . . .	751
8.730.2.4 failureReasonv6 . . . . .	751
8.730.2.5 instanceId . . . . .	751
8.730.2.6 ipfamily . . . . .	751
8.730.2.7 pAuthentication . . . . .	751
8.730.2.8 pPassword . . . . .	751
8.730.2.9 pProfileId3GPP . . . . .	751
8.730.2.10pProfileId3GPP2 . . . . .	751
8.730.2.11pTechnology . . . . .	752

8.730.2.12pUsername . . . . .	752
8.730.2.13cv4 . . . . .	752
8.730.2.14cv6 . . . . .	752
8.730.2.15sessionId . . . . .	752
8.730.2.16v4sessionId . . . . .	752
8.730.2.17v6sessionId . . . . .	752
8.730.2.18verbFailReason . . . . .	752
8.730.2.19verbFailReasonType . . . . .	752
8.731 SupportedMsgList Struct Reference . . . . .	752
8.731.1 Detailed Description . . . . .	752
8.731.2 Field Documentation . . . . .	752
8.731.2.1 supportedMsgLen . . . . .	752
8.731.2.2 supportedMsgs . . . . .	752
8.732 SUPSInfo Struct Reference . . . . .	752
8.732.1 Detailed Description . . . . .	753
8.732.2 Field Documentation . . . . .	753
8.732.2.1 isModByCC . . . . .	753
8.732.2.2 svcType . . . . .	753
8.733 SV Struct Reference . . . . .	753
8.733.1 Detailed Description . . . . .	753
8.733.2 Field Documentation . . . . .	754
8.733.2.1 id . . . . .	754
8.733.2.2 mask . . . . .	754
8.733.2.3 system . . . . .	754
8.734 SVInfo Struct Reference . . . . .	754
8.734.1 Detailed Description . . . . .	754
8.734.2 Field Documentation . . . . .	754
8.734.2.1 len . . . . .	755
8.734.2.2 pSV . . . . .	755
8.735 svUsedforFix_s Struct Reference . . . . .	755
8.735.1 Detailed Description . . . . .	755
8.735.2 Field Documentation . . . . .	755
8.735.2.1 gnssSvUsedList . . . . .	755
8.735.2.2 gnssSvUsedList_len . . . . .	755
8.736 SWI_STRUCT_CarrierImage Struct Reference . . . . .	755
8.736.1 Detailed Description . . . . .	755
8.736.2 Field Documentation . . . . .	756
8.736.2.1 m_FwBuildId . . . . .	756
8.736.2.2 m_FwImageld . . . . .	756
8.736.2.3 m_nCarrierId . . . . .	756

8.736.2.4 m_nFolderId . . . . .	756
8.736.2.5 m_nStorage . . . . .	756
8.736.2.6 m_PriBuildId . . . . .	756
8.736.2.7 m_PrImagId . . . . .	756
8.737SwiLocGetAutoStartResp Struct Reference . . . . .	756
8.737.1 Detailed Description . . . . .	757
8.737.2 Field Documentation . . . . .	758
8.737.2.1 fix_rate . . . . .	758
8.737.2.2 fix_rate_reported . . . . .	758
8.737.2.3 fix_type . . . . .	758
8.737.2.4 fix_type_reported . . . . .	758
8.737.2.5 function . . . . .	758
8.737.2.6 function_reported . . . . .	758
8.737.2.7 max_dist . . . . .	758
8.737.2.8 max_dist_reported . . . . .	758
8.737.2.9 max_time . . . . .	758
8.737.2.10max_time_reported . . . . .	758
8.738SwiLocSetAutoStartReq Struct Reference . . . . .	758
8.738.1 Detailed Description . . . . .	758
8.738.2 Field Documentation . . . . .	759
8.738.2.1 fix_rate . . . . .	759
8.738.2.2 fix_type . . . . .	759
8.738.2.3 function . . . . .	759
8.738.2.4 max_dist . . . . .	759
8.738.2.5 max_time . . . . .	759
8.738.2.6 set_fix_rate . . . . .	760
8.738.2.7 set_fix_type . . . . .	760
8.738.2.8 set_function . . . . .	760
8.738.2.9 set_max_dist . . . . .	760
8.738.2.10set_max_time . . . . .	760
8.739swiModemStatusResp Struct Reference . . . . .	760
8.739.1 Detailed Description . . . . .	760
8.739.2 Field Documentation . . . . .	760
8.739.2.1 commonInfo . . . . .	760
8.739.2.2 pLTEInfo . . . . .	760
8.740SwiOTAMsg_s Struct Reference . . . . .	760
8.740.1 Detailed Description . . . . .	760
8.740.2 Field Documentation . . . . .	761
8.740.2.1 data . . . . .	761
8.740.2.2 data_len . . . . .	761

8.740.2.3 pLteNasRelInfo . . . . .	761
8.740.2.4 pTime . . . . .	761
8.740.2.5 type . . . . .	761
8.741swiPDPRuntimeSettingsReq Struct Reference . . . . .	761
8.741.1 Detailed Description . . . . .	761
8.741.2 Field Documentation . . . . .	762
8.741.2.1 contextId . . . . .	762
8.741.2.2 contextType . . . . .	762
8.742swiPDPRuntimeSettingsResp Struct Reference . . . . .	762
8.742.1 Detailed Description . . . . .	762
8.742.2 Field Documentation . . . . .	764
8.742.2.1 pAPNName . . . . .	764
8.742.2.2 pBearerId . . . . .	764
8.742.2.3 pContextId . . . . .	764
8.742.2.4 pIPv4Address . . . . .	764
8.742.2.5 pIPv4GWAddress . . . . .	764
8.742.2.6 pIPv6Address . . . . .	764
8.742.2.7 pIPv6GWAddress . . . . .	764
8.742.2.8 pPrDNSIPv4Address . . . . .	764
8.742.2.9 pPrDNSIPv6Address . . . . .	764
8.742.2.10pPrPCSCFIPv4Address . . . . .	764
8.742.2.11pPrPCSCFIPv6Address . . . . .	764
8.742.2.12pSeDNSIPv4Address . . . . .	764
8.742.2.13pSeDNSIPv6Address . . . . .	764
8.742.2.14pSePCSCFIPv4Address . . . . .	764
8.742.2.15pSePCSCFIPv6Address . . . . .	764
8.743swiQosFilter Struct Reference . . . . .	764
8.743.1 Detailed Description . . . . .	765
8.743.2 Field Documentation . . . . .	766
8.743.2.1 index . . . . .	766
8.743.2.2 pEspSpi . . . . .	766
8.743.2.3 pId . . . . .	766
8.743.2.4 pIPv4DstAddr . . . . .	766
8.743.2.5 pIPv4SrcAddr . . . . .	766
8.743.2.6 pIPv6DstAddr . . . . .	766
8.743.2.7 pIPv6Label . . . . .	766
8.743.2.8 pIPv6SrcAddr . . . . .	766
8.743.2.9 pIPv6TrafCls . . . . .	766
8.743.2.10pNxtHdrProto . . . . .	766
8.743.2.11pPrecedence . . . . .	766

8.743.2.12pTCPDstPort . . . . .	766
8.743.2.13pTCPSrcPort . . . . .	766
8.743.2.14pTos . . . . .	766
8.743.2.15pTranDstPort . . . . .	766
8.743.2.16pTranSrcPort . . . . .	766
8.743.2.17pUDPDstPort . . . . .	766
8.743.2.18pUDPSrcPort . . . . .	767
8.743.2.19version . . . . .	767
8.744swiQosFlow Struct Reference . . . . .	767
8.744.1 Detailed Description . . . . .	767
8.744.2 Field Documentation . . . . .	769
8.744.2.1 index . . . . .	769
8.744.2.2 p3GPP2Pri . . . . .	769
8.744.2.3 p3GPPImCn . . . . .	769
8.744.2.4 p3GPPResResidualBER . . . . .	769
8.744.2.5 p3GPPSigInd . . . . .	769
8.744.2.6 p3GPPTraHdlPri . . . . .	769
8.744.2.7 pDataRate . . . . .	769
8.744.2.8 pJitter . . . . .	769
8.744.2.9 pLatency . . . . .	769
8.744.2.10pLteQci . . . . .	769
8.744.2.11pMaxAllowedPktSz . . . . .	770
8.744.2.12pMinPolicedPktSz . . . . .	770
8.744.2.13pPktErrRate . . . . .	770
8.744.2.14pProfileId3GPP2 . . . . .	770
8.744.2.15pTokenBucket . . . . .	770
8.744.2.16pTrafficClass . . . . .	770
8.745swiQosGranted Struct Reference . . . . .	770
8.745.1 Detailed Description . . . . .	770
8.745.2 Field Documentation . . . . .	770
8.745.2.1 pRxFlow . . . . .	770
8.745.2.2 pTxFlow . . . . .	770
8.746swiQosIds Struct Reference . . . . .	770
8.746.1 Detailed Description . . . . .	770
8.746.2 Field Documentation . . . . .	770
8.746.2.1 plds . . . . .	771
8.746.2.2 sz . . . . .	771
8.747swiQosModifyReq Struct Reference . . . . .	771
8.747.1 Detailed Description . . . . .	771
8.747.2 Field Documentation . . . . .	771

8.747.2.1 id	771
8.747.2.2 pRxFilter	771
8.747.2.3 pRxFlow	771
8.747.2.4 pTxFilter	771
8.747.2.5 pTxFlow	771
8.748swiQosReq Struct Reference	771
8.748.1 Detailed Description	771
8.748.2 Field Documentation	772
8.748.2.1 index	772
8.748.2.2 pRxFilter	772
8.748.2.3 pRxFlow	772
8.748.2.4 pTxFilter	772
8.748.2.5 pTxFlow	772
8.749swiRMTrasnferStaticsReq Struct Reference	772
8.749.1 Detailed Description	772
8.749.2 Field Documentation	773
8.749.2.1 bResetStatistics	773
8.749.2.2 ulMask	773
8.750sysInfoCommon Struct Reference	773
8.750.1 Detailed Description	773
8.750.2 Field Documentation	775
8.750.2.1 isSysForbidden	775
8.750.2.2 isSysForbiddenValid	775
8.750.2.3 roamStatus	775
8.750.2.4 roamStatusValid	775
8.750.2.5 srvCapability	775
8.750.2.6 srvCapabilityValid	775
8.750.2.7 srvDomain	775
8.750.2.8 srvDomainValid	775
8.751t_gpsTime Struct Reference	775
8.751.1 Field Documentation	775
8.751.1.1 gpsTimeOfWeekMs	775
8.751.1.2 gpsWeek	775
8.752t_sensor Struct Reference	775
8.752.1 Field Documentation	775
8.752.1.1 aidingIndicatorMask	775
8.752.1.2 usageMask	775
8.753t_Sv Struct Reference	776
8.753.1 Field Documentation	776
8.753.1.1 entries	776

8.753.1.2 len . . . . .	776
8.754TDSCDMAECIOThresh Struct Reference . . . . .	776
8.754.1 Detailed Description . . . . .	776
8.754.2 Field Documentation . . . . .	776
8.754.2.1 pTDSCDMAECIOThreshList . . . . .	776
8.754.2.2 TDSCDMAECIOThreshListLen . . . . .	776
8.755TDSCDMARSCPTthresh Struct Reference . . . . .	776
8.755.1 Detailed Description . . . . .	776
8.755.2 Field Documentation . . . . .	777
8.755.2.1 pTDSCDMARSCPTthreshList . . . . .	777
8.755.2.2 TDSCDMARSCPTthreshListLen . . . . .	777
8.756TDSCDMARSSIThresh Struct Reference . . . . .	777
8.756.1 Detailed Description . . . . .	777
8.756.2 Field Documentation . . . . .	777
8.756.2.1 pTDSCDMARSSIThreshList . . . . .	777
8.756.2.2 TDSCDMARSSIThreshListLen . . . . .	777
8.757TDSCDMASigInfoExt Struct Reference . . . . .	777
8.757.1 Detailed Description . . . . .	778
8.757.2 Field Documentation . . . . .	778
8.757.2.1 ecio . . . . .	778
8.757.2.2 rscp . . . . .	778
8.757.2.3 rssi . . . . .	778
8.757.2.4 sinr . . . . .	778
8.758tdscdmaSigInfoExt Struct Reference . . . . .	778
8.758.1 Detailed Description . . . . .	778
8.758.2 Field Documentation . . . . .	778
8.758.2.1 ecio . . . . .	778
8.758.2.2 rscp . . . . .	778
8.758.2.3 rssi . . . . .	779
8.758.2.4 sinr . . . . .	779
8.759TDSCDMASINRCONFThresh Struct Reference . . . . .	779
8.759.1 Detailed Description . . . . .	779
8.759.2 Field Documentation . . . . .	779
8.759.2.1 pTDSCDMASINRCONFThreshList . . . . .	779
8.759.2.2 TDSCDMASINRCONFThreshListLen . . . . .	779
8.760TDSCDMASINRThresh Struct Reference . . . . .	779
8.760.1 Detailed Description . . . . .	779
8.760.2 Field Documentation . . . . .	780
8.760.2.1 pTDSCDMASINRThreshList . . . . .	780
8.760.2.2 TDSCDMASINRThreshListLen . . . . .	780

8.761temperatureData Struct Reference . . . . .	780
8.761.1 Detailed Description . . . . .	780
8.761.2 Field Documentation . . . . .	780
8.761.2.1 temperature . . . . .	780
8.761.2.2 temperatureDataLen . . . . .	781
8.761.2.3 timeOfFirstSample . . . . .	781
8.761.2.4 timeOffset . . . . .	781
8.761.2.5 timeSource . . . . .	781
8.762TFTIDParams Struct Reference . . . . .	781
8.762.1 Detailed Description . . . . .	781
8.762.2 Field Documentation . . . . .	782
8.762.2.1 destPortRangeEnd . . . . .	782
8.762.2.2 destPortRangeStart . . . . .	782
8.762.2.3 eValid . . . . .	782
8.762.2.4 filterId . . . . .	782
8.762.2.5 flowLabel . . . . .	782
8.762.2.6 IPSECSPi . . . . .	782
8.762.2.7 ipVersion . . . . .	782
8.762.2.8 nextHeader . . . . .	782
8.762.2.9 pSourceIP . . . . .	782
8.762.2.10sourceIPMask . . . . .	782
8.762.2.11srcPortRangeEnd . . . . .	782
8.762.2.12srcPortRangeStart . . . . .	782
8.762.2.13osMask . . . . .	782
8.763timeInfo Struct Reference . . . . .	783
8.763.1 Detailed Description . . . . .	783
8.763.2 Field Documentation . . . . .	784
8.763.2.1 day . . . . .	784
8.763.2.2 dayLtSavingAdj . . . . .	784
8.763.2.3 dayOfWeek . . . . .	784
8.763.2.4 hour . . . . .	784
8.763.2.5 minute . . . . .	784
8.763.2.6 month . . . . .	784
8.763.2.7 radioInterface . . . . .	784
8.763.2.8 second . . . . .	784
8.763.2.9 timeZone . . . . .	784
8.763.2.10TlvPresent . . . . .	784
8.763.2.11year . . . . .	784
8.764TmdDeRegNotMitigationLvIReq Struct Reference . . . . .	784
8.764.1 Detailed Description . . . . .	785

8.764.2 Field Documentation . . . . .	785
8.764.2.1 mitigationDevID . . . . .	785
8.764.2.2 mitigationDevIDLen . . . . .	785
8.765TmdGetMitigationDevListResp Struct Reference . . . . .	785
8.765.1 Detailed Description . . . . .	785
8.765.2 Field Documentation . . . . .	785
8.765.2.1 pMitigationDevList . . . . .	785
8.765.2.2 pMitigationDevListLen . . . . .	785
8.766TmdGetMitigationLvlReq Struct Reference . . . . .	785
8.766.1 Detailed Description . . . . .	786
8.766.2 Field Documentation . . . . .	786
8.766.2.1 mitigationDevID . . . . .	786
8.766.2.2 mitigationDevIDLen . . . . .	786
8.767TmdGetMitigationLvlResp Struct Reference . . . . .	786
8.767.1 Detailed Description . . . . .	786
8.767.2 Field Documentation . . . . .	786
8.767.2.1 pCurrentmitigationLvl . . . . .	786
8.767.2.2 pReqMitigationLvl . . . . .	786
8.768TmdMitigationLvlIndReq Struct Reference . . . . .	787
8.768.1 Detailed Description . . . . .	787
8.768.2 Field Documentation . . . . .	787
8.768.2.1 mitigationDevID . . . . .	787
8.768.2.2 mitigationDevIDLen . . . . .	787
8.769TmdRegNotMitigationLvlReq Struct Reference . . . . .	787
8.769.1 Detailed Description . . . . .	787
8.769.2 Field Documentation . . . . .	787
8.769.2.1 mitigationDevID . . . . .	787
8.769.2.2 mitigationDevIDLen . . . . .	788
8.770tokenBucket Struct Reference . . . . .	788
8.770.1 Detailed Description . . . . .	788
8.770.2 Field Documentation . . . . .	788
8.770.2.1 bucketSz . . . . .	788
8.770.2.2 peakRate . . . . .	788
8.770.2.3 tokenRate . . . . .	788
8.771Tos Struct Reference . . . . .	788
8.771.1 Detailed Description . . . . .	788
8.771.2 Field Documentation . . . . .	789
8.771.2.1 mask . . . . .	789
8.771.2.2 val . . . . .	789
8.772transferRouteMessageTlv Struct Reference . . . . .	789

8.772.1 Detailed Description . . . . .	789
8.772.2 Field Documentation . . . . .	789
8.772.2.1 TlvPresent . . . . .	789
8.772.2.2 TransferRouteMTMessageInfo . . . . .	789
8.773TransferStatInd Struct Reference . . . . .	789
8.773.1 Detailed Description . . . . .	789
8.773.2 Field Documentation . . . . .	790
8.773.2.1 StatsMask . . . . .	790
8.773.2.2 StatsPeriod . . . . .	790
8.774transferStatInd Struct Reference . . . . .	790
8.774.1 Detailed Description . . . . .	790
8.774.2 Field Documentation . . . . .	790
8.774.2.1 StatsMask . . . . .	790
8.774.2.2 StatsPeriod . . . . .	790
8.775TransferStatsDataType Struct Reference . . . . .	790
8.775.1 Field Documentation . . . . .	790
8.775.1.1 interval . . . . .	790
8.776TrStatInd Struct Reference . . . . .	790
8.776.1 Detailed Description . . . . .	791
8.776.2 Field Documentation . . . . .	791
8.776.2.1 statsMask . . . . .	791
8.776.2.2 statsPeriod . . . . .	791
8.777trueIMSI Struct Reference . . . . .	791
8.777.1 Detailed Description . . . . .	791
8.777.2 Field Documentation . . . . .	792
8.777.2.1 imsiT1112 . . . . .	792
8.777.2.2 imsiTaddrNum . . . . .	792
8.777.2.3 imsiTS1 . . . . .	792
8.777.2.4 imsiTS2 . . . . .	792
8.777.2.5 mccT . . . . .	792
8.778TXAGCList Struct Reference . . . . .	792
8.778.1 Detailed Description . . . . .	792
8.778.2 Field Documentation . . . . .	793
8.778.2.1 pTXAIG . . . . .	793
8.778.2.2 pTXComprSlope . . . . .	793
8.778.2.3 pTXComprThres . . . . .	793
8.778.2.4 pTXExpSlope . . . . .	793
8.778.2.5 pTXExpThres . . . . .	793
8.778.2.6 pTXStaticGain . . . . .	793
8.779txInfo Struct Reference . . . . .	793

8.779.1 Detailed Description . . . . .	793
8.779.2 Field Documentation . . . . .	793
8.779.2.1 isInTraffic . . . . .	793
8.779.2.2 txPower . . . . .	794
8.780TXPCMIIRFtr Struct Reference . . . . .	794
8.780.1 Detailed Description . . . . .	794
8.780.2 Field Documentation . . . . .	795
8.780.2.1 pFlag . . . . .	795
8.780.2.2 pStage0Val . . . . .	795
8.780.2.3 pStage1Val . . . . .	795
8.780.2.4 pStage2Val . . . . .	795
8.780.2.5 pStage3Val . . . . .	795
8.780.2.6 pStage4Val . . . . .	795
8.780.2.7 pStageCnt . . . . .	795
8.781uim_appStatus Struct Reference . . . . .	795
8.781.1 Detailed Description . . . . .	796
8.781.2 Field Documentation . . . . .	797
8.781.2.1 aidLength . . . . .	798
8.781.2.2 aidVal . . . . .	798
8.781.2.3 appState . . . . .	798
8.781.2.4 appType . . . . .	798
8.781.2.5 persoFeature . . . . .	798
8.781.2.6 persoRetries . . . . .	798
8.781.2.7 persoState . . . . .	798
8.781.2.8 persoUnblockRetries . . . . .	798
8.781.2.9 pin1Retries . . . . .	798
8.781.2.10pin1State . . . . .	798
8.781.2.11pin2Retries . . . . .	798
8.781.2.12pin2State . . . . .	798
8.781.2.13puk1Retries . . . . .	798
8.781.2.14puk2Retries . . . . .	798
8.781.2.15univPin . . . . .	798
8.782uim_cardResult Struct Reference . . . . .	798
8.782.1 Detailed Description . . . . .	798
8.782.2 Field Documentation . . . . .	798
8.782.2.1 sw1 . . . . .	798
8.782.2.2 sw2 . . . . .	799
8.783uim_cardStatus Struct Reference . . . . .	799
8.783.1 Detailed Description . . . . .	799
8.783.2 Field Documentation . . . . .	800

8.783.2.1 index1xPri . . . . .	800
8.783.2.2 index1xSec . . . . .	800
8.783.2.3 indexGwPri . . . . .	800
8.783.2.4 indexGwSec . . . . .	800
8.783.2.5 numSlot . . . . .	800
8.783.2.6 SlotInfo . . . . .	800
8.784uim_changeUIMPIN Struct Reference . . . . .	800
8.784.1 Detailed Description . . . . .	800
8.784.2 Field Documentation . . . . .	800
8.784.2.1 oldPINLen . . . . .	801
8.784.2.2 oldPINVal . . . . .	801
8.784.2.3 pinID . . . . .	801
8.784.2.4 pinLen . . . . .	801
8.784.2.5 pinVal . . . . .	801
8.785uim_encryptedPIN1 Struct Reference . . . . .	801
8.785.1 Detailed Description . . . . .	801
8.785.2 Field Documentation . . . . .	801
8.785.2.1 pin1Len . . . . .	801
8.785.2.2 pin1Val . . . . .	801
8.786uim_fileInfo Struct Reference . . . . .	801
8.786.1 Detailed Description . . . . .	801
8.786.2 Field Documentation . . . . .	802
8.786.2.1 fileID . . . . .	802
8.786.2.2 path . . . . .	802
8.786.2.3 pathLen . . . . .	802
8.787uim_hotSwapStatus Struct Reference . . . . .	802
8.787.1 Detailed Description . . . . .	802
8.787.2 Field Documentation . . . . .	802
8.787.2.1 hotSwap . . . . .	802
8.787.2.2 hotSwapLength . . . . .	802
8.788uim_readResult Struct Reference . . . . .	802
8.788.1 Detailed Description . . . . .	803
8.788.2 Field Documentation . . . . .	803
8.788.2.1 content . . . . .	803
8.788.2.2 contentLen . . . . .	803
8.789uim_readTransparentInfo Struct Reference . . . . .	803
8.789.1 Detailed Description . . . . .	803
8.789.2 Field Documentation . . . . .	803
8.789.2.1 length . . . . .	803
8.789.2.2 offset . . . . .	803

8.790uim_remainingRetries Struct Reference . . . . .	804
8.790.1 Detailed Description . . . . .	804
8.790.2 Field Documentation . . . . .	804
8.790.2.1 unblockLeft . . . . .	804
8.790.2.2 verifyLeft . . . . .	804
8.791uim_sessionInformation Struct Reference . . . . .	804
8.791.1 Detailed Description . . . . .	804
8.791.2 Field Documentation . . . . .	805
8.791.2.1 aid . . . . .	805
8.791.2.2 aidLength . . . . .	805
8.791.2.3 sessionType . . . . .	805
8.792uim_setPINProtection Struct Reference . . . . .	805
8.792.1 Detailed Description . . . . .	805
8.792.2 Field Documentation . . . . .	806
8.792.2.1 pinID . . . . .	806
8.792.2.2 pinLength . . . . .	806
8.792.2.3 pinOperation . . . . .	806
8.792.2.4 pinValue . . . . .	806
8.793uim_slotInfo Struct Reference . . . . .	806
8.793.1 Detailed Description . . . . .	806
8.793.2 Field Documentation . . . . .	807
8.793.2.1 AppStatus . . . . .	807
8.793.2.2 cardState . . . . .	807
8.793.2.3 errorState . . . . .	807
8.793.2.4 numApp . . . . .	807
8.793.2.5 upinRetries . . . . .	808
8.793.2.6 upinState . . . . .	808
8.793.2.7 upukRetries . . . . .	808
8.794uim_UIMSessionInformation Struct Reference . . . . .	808
8.794.1 Detailed Description . . . . .	808
8.794.2 Field Documentation . . . . .	808
8.794.2.1 aid . . . . .	808
8.794.2.2 aidLength . . . . .	808
8.794.2.3 sessionType . . . . .	808
8.795uim_unblockUIMPIN Struct Reference . . . . .	808
8.795.1 Detailed Description . . . . .	809
8.795.2 Field Documentation . . . . .	809
8.795.2.1 newPINLen . . . . .	809
8.795.2.2 newPINVal . . . . .	809
8.795.2.3 pinID . . . . .	809

8.795.2.4 pukLen . . . . .	809
8.795.2.5 pukVal . . . . .	809
8.796uim_verifyUIMPIN Struct Reference . . . . .	809
8.796.1 Detailed Description . . . . .	810
8.796.2 Field Documentation . . . . .	810
8.796.2.1 pinID . . . . .	810
8.796.2.2 pinLen . . . . .	810
8.796.2.3 pinVal . . . . .	810
8.797UIMAuthenticateReq Struct Reference . . . . .	810
8.797.1 Detailed Description . . . . .	810
8.797.2 Field Documentation . . . . .	811
8.797.2.1 authData . . . . .	811
8.797.2.2 pIndicationToken . . . . .	811
8.797.2.3 sessionInfo . . . . .	811
8.798UIMAuthenticateResp Struct Reference . . . . .	811
8.798.1 Detailed Description . . . . .	811
8.798.2 Field Documentation . . . . .	811
8.798.2.1 pAuthenticateResult . . . . .	811
8.798.2.2 pCardResult . . . . .	811
8.798.2.3 pIndicationToken . . . . .	812
8.799UIMChangePinReq Struct Reference . . . . .	812
8.799.1 Detailed Description . . . . .	812
8.799.2 Field Documentation . . . . .	812
8.799.2.1 changePIN . . . . .	812
8.799.2.2 pIndicationToken . . . . .	812
8.799.2.3 pKeyReferenceID . . . . .	812
8.799.2.4 sessionInfo . . . . .	812
8.800UIMDepersonalizationReq Struct Reference . . . . .	812
8.800.1 Detailed Description . . . . .	813
8.800.2 Field Documentation . . . . .	813
8.800.2.1 depersonilisationInfo . . . . .	813
8.801UIMDepersonalizationResp Struct Reference . . . . .	813
8.801.1 Detailed Description . . . . .	813
8.801.2 Field Documentation . . . . .	813
8.801.2.1 pRemainingRetries . . . . .	813
8.802UIMEventRegisterReqResp Struct Reference . . . . .	813
8.802.1 Detailed Description . . . . .	813
8.802.2 Field Documentation . . . . .	814
8.802.2.1 eventMask . . . . .	814
8.803UIMGetCardStatusResp Struct Reference . . . . .	814

8.803.1 Detailed Description . . . . .	814
8.803.2 Field Documentation . . . . .	814
8.803.2.1 pCardStatus . . . . .	814
8.803.2.2 pHotSwapStatus . . . . .	814
8.804UIMGetConfigurationReq Struct Reference . . . . .	814
8.804.1 Detailed Description . . . . .	815
8.804.2 Field Documentation . . . . .	815
8.804.2.1 pConfigurationMask . . . . .	815
8.805UIMGetConfigurationResp Struct Reference . . . . .	815
8.805.1 Detailed Description . . . . .	815
8.805.2 Field Documentation . . . . .	816
8.805.2.1 pAutoSelection . . . . .	816
8.805.2.2 pHaltSubscription . . . . .	816
8.805.2.3 pPersonalizationStatus . . . . .	816
8.806UIMGetFileAttributesReq Struct Reference . . . . .	816
8.806.1 Detailed Description . . . . .	816
8.806.2 Field Documentation . . . . .	816
8.806.2.1 fileIndex . . . . .	816
8.806.2.2 pIndicationToken . . . . .	816
8.806.2.3 sessionInfo . . . . .	816
8.807UIMGetFileAttributesResp Struct Reference . . . . .	816
8.807.1 Detailed Description . . . . .	817
8.807.2 Field Documentation . . . . .	817
8.807.2.1 pCardResult . . . . .	817
8.807.2.2 pFileAttributes . . . . .	817
8.807.2.3 pIndicationToken . . . . .	817
8.808UIMGetSlotsStatusResp Struct Reference . . . . .	817
8.808.1 Detailed Description . . . . .	817
8.808.2 Field Documentation . . . . .	817
8.808.2.1 pNumberOfPhySlot . . . . .	817
8.808.2.2 pUimSlotsStatus . . . . .	818
8.809UIMPinResp Struct Reference . . . . .	818
8.809.1 Detailed Description . . . . .	818
8.809.2 Field Documentation . . . . .	818
8.809.2.1 pEncryptedPIN1 . . . . .	818
8.809.2.2 pIndicationToken . . . . .	818
8.809.2.3 pRemainingRetries . . . . .	818
8.810UIMPowerDownReq Struct Reference . . . . .	818
8.810.1 Detailed Description . . . . .	818
8.810.2 Field Documentation . . . . .	819

8.810.2.1 slot . . . . .	819
8.811UIMPowerUpReq Struct Reference . . . . .	819
8.811.1 Detailed Description . . . . .	819
8.811.2 Field Documentation . . . . .	819
8.811.2.1 plgnoreHotSwapSwitch . . . . .	819
8.811.2.2 slot . . . . .	819
8.812UIMReadTransparentReq Struct Reference . . . . .	819
8.812.1 Detailed Description . . . . .	820
8.812.2 Field Documentation . . . . .	820
8.812.2.1 fileIndex . . . . .	820
8.812.2.2 pEncryptData . . . . .	820
8.812.2.3 plndicationToken . . . . .	820
8.812.2.4 readTransparent . . . . .	820
8.812.2.5 sessionInfo . . . . .	820
8.813UIMReadTransparentResp Struct Reference . . . . .	820
8.813.1 Detailed Description . . . . .	820
8.813.2 Field Documentation . . . . .	821
8.813.2.1 pCardResult . . . . .	821
8.813.2.2 pEncryptedData . . . . .	821
8.813.2.3 plndicationToken . . . . .	821
8.813.2.4 pReadResult . . . . .	821
8.814UIMRefreshCompleteReq Struct Reference . . . . .	821
8.814.1 Detailed Description . . . . .	821
8.814.2 Field Documentation . . . . .	822
8.814.2.1 refreshComplete . . . . .	822
8.814.2.2 sessionInfo . . . . .	822
8.815UIMRefreshEvent Struct Reference . . . . .	822
8.815.1 Detailed Description . . . . .	822
8.815.2 Field Documentation . . . . .	823
8.815.2.1 aid . . . . .	823
8.815.2.2 aidLength . . . . .	823
8.815.2.3 arrfileInfo . . . . .	823
8.815.2.4 mode . . . . .	823
8.815.2.5 numOfFiles . . . . .	823
8.815.2.6 sessionType . . . . .	823
8.815.2.7 stage . . . . .	823
8.816UIMRefreshGetLastEventReq Struct Reference . . . . .	823
8.816.1 Detailed Description . . . . .	824
8.816.2 Field Documentation . . . . .	824
8.816.2.1 sessionInfo . . . . .	824

8.817UIMRefreshGetLastEventResp Struct Reference . . . . .	824
8.817.1 Detailed Description . . . . .	824
8.817.2 Field Documentation . . . . .	824
8.817.2.1 pRefreshEvent . . . . .	824
8.818UIMRefreshOKReq Struct Reference . . . . .	824
8.818.1 Detailed Description . . . . .	824
8.818.2 Field Documentation . . . . .	825
8.818.2.1 OKtoRefresh . . . . .	825
8.818.2.2 sessionInfo . . . . .	825
8.819UIMRefreshRegisterReq Struct Reference . . . . .	825
8.819.1 Detailed Description . . . . .	825
8.819.2 Field Documentation . . . . .	825
8.819.2.1 regRefresh . . . . .	825
8.819.2.2 sessionInfo . . . . .	825
8.820UIMSessionInformation Struct Reference . . . . .	825
8.820.1 Detailed Description . . . . .	826
8.820.2 Field Documentation . . . . .	826
8.820.2.1 aid . . . . .	826
8.820.2.2 aidLength . . . . .	826
8.820.2.3 sessionType . . . . .	826
8.821UIMSetPinProtectionReq Struct Reference . . . . .	826
8.821.1 Detailed Description . . . . .	826
8.821.2 Field Documentation . . . . .	827
8.821.2.1 pIndicationToken . . . . .	827
8.821.2.2 pinProtection . . . . .	827
8.821.2.3 pKeyReferenceID . . . . .	827
8.821.2.4 sessionInfo . . . . .	827
8.822UIMSlotsStatus Struct Reference . . . . .	827
8.822.1 Detailed Description . . . . .	827
8.822.2 Field Documentation . . . . .	827
8.822.2.1 uimSlotStatus . . . . .	827
8.823UIMSlotStatus Struct Reference . . . . .	827
8.823.1 Detailed Description . . . . .	828
8.823.2 Field Documentation . . . . .	828
8.823.2.1 bICCID . . . . .	828
8.823.2.2 bICCIDLength . . . . .	828
8.823.2.3 bLogicalSlot . . . . .	828
8.823.2.4 uPhyCardStatus . . . . .	828
8.823.2.5 uPhySlotStatus . . . . .	828
8.824UIMSlotStatusChangeInfo Struct Reference . . . . .	829

8.824.1 Detailed Description . . . . .	829
8.824.2 Field Documentation . . . . .	829
8.824.2.1 bNumberOfPhySlots . . . . .	829
8.824.2.2 slotsstatusChange . . . . .	829
8.825UIMStatusChangeInfo Struct Reference . . . . .	829
8.825.1 Detailed Description . . . . .	829
8.825.2 Field Documentation . . . . .	829
8.825.2.1 statusChange . . . . .	829
8.826UIMSwitchSlotReq Struct Reference . . . . .	829
8.826.1 Detailed Description . . . . .	830
8.826.2 Field Documentation . . . . .	830
8.826.2.1 bLogicalSlot . . . . .	830
8.826.2.2 ulPhysicalSlot . . . . .	830
8.827UIMUnblockPinReq Struct Reference . . . . .	830
8.827.1 Detailed Description . . . . .	830
8.827.2 Field Documentation . . . . .	831
8.827.2.1 pIndicationToken . . . . .	831
8.827.2.2 pKeyReferenceID . . . . .	831
8.827.2.3 sessionInfo . . . . .	831
8.827.2.4 unblockPIN . . . . .	831
8.828UIMVerifyPinReq Struct Reference . . . . .	831
8.828.1 Detailed Description . . . . .	831
8.828.2 Field Documentation . . . . .	832
8.828.2.1 pEncryptedPIN1 . . . . .	832
8.828.2.2 pIndicationToken . . . . .	832
8.828.2.3 pKeyReferenceID . . . . .	832
8.828.2.4 sessionInfo . . . . .	832
8.828.2.5 verifyPIN . . . . .	832
8.829UMTSInfo Struct Reference . . . . .	832
8.829.1 Detailed Description . . . . .	832
8.829.2 Field Documentation . . . . .	833
8.829.2.1 cellID . . . . .	833
8.829.2.2 ecio . . . . .	833
8.829.2.3 geranInst . . . . .	833
8.829.2.4 GeranInstInfo . . . . .	833
8.829.2.5 lac . . . . .	833
8.829.2.6 plmn . . . . .	834
8.829.2.7 psc . . . . .	834
8.829.2.8 rscp . . . . .	834
8.829.2.9 uarfcn . . . . .	834

8.829.2.10	umtsInst	834
8.829.2.11	UMTSInstInfo	834
8.830	UMTSInstInfo Struct Reference	834
8.830.1	Detailed Description	834
8.830.2	Field Documentation	834
8.830.2.1	umtsEcio	834
8.830.2.2	umtsPsc	834
8.830.2.3	umtsRscp	834
8.830.2.4	umtsUarfcn	834
8.831	umtsLTENbrCell Struct Reference	834
8.831.1	Detailed Description	835
8.831.2	Field Documentation	835
8.831.2.1	cellsTDD	835
8.831.2.2	earfcn	835
8.831.2.3	pci	835
8.831.2.4	rsrp	835
8.831.2.5	rsrq	835
8.831.2.6	srxlev	835
8.832	UMTSMinQoS Struct Reference	835
8.832.1	Detailed Description	836
8.832.2	Field Documentation	837
8.832.2.1	deliveryErrSDU	837
8.832.2.2	grntDownlinkBitrate	838
8.832.2.3	grntUplinkBitrate	838
8.832.2.4	maxDownlinkBitrate	838
8.832.2.5	maxSDUSize	838
8.832.2.6	maxUplinkBitrate	838
8.832.2.7	qosDeliveryOrder	838
8.832.2.8	resBerRatio	838
8.832.2.9	sduErrorRatio	838
8.832.2.10	trafficClass	838
8.832.2.11	trafficPriority	838
8.832.2.12	transferDelay	838
8.833	UMTSQoS Struct Reference	838
8.833.1	Detailed Description	838
8.833.2	Field Documentation	840
8.833.2.1	deliveryErrSDU	840
8.833.2.2	grntDownlinkBitrate	840
8.833.2.3	grntUplinkBitrate	840
8.833.2.4	maxDownlinkBitrate	840

8.833.2.5 maxSDUSize . . . . .	840
8.833.2.6 maxUplinkBitrate . . . . .	840
8.833.2.7 qosDeliveryOrder . . . . .	840
8.833.2.8 resBerRatio . . . . .	840
8.833.2.9 sduErrorRatio . . . . .	840
8.833.2.10 trafficClass . . . . .	840
8.833.2.11 trafficPriority . . . . .	840
8.833.2.12 transferDelay . . . . .	840
8.834 UMTSReqQoS SigInd Struct Reference . . . . .	840
8.834.1 Detailed Description . . . . .	841
8.834.2 Field Documentation . . . . .	841
8.834.2.1 SigInd . . . . .	841
8.834.2.2 UMTSReqQoS . . . . .	841
8.835 unblockUIMPIN Struct Reference . . . . .	841
8.835.1 Detailed Description . . . . .	841
8.835.2 Field Documentation . . . . .	842
8.835.2.1 newPINLen . . . . .	842
8.835.2.2 newPINVal . . . . .	842
8.835.2.3 pinID . . . . .	842
8.835.2.4 pukLen . . . . .	842
8.835.2.5 pukVal . . . . .	842
8.836 UniversalTime Struct Reference . . . . .	842
8.836.1 Detailed Description . . . . .	842
8.836.2 Field Documentation . . . . .	843
8.836.2.1 day . . . . .	843
8.836.2.2 dayOfWeek . . . . .	843
8.836.2.3 hour . . . . .	843
8.836.2.4 minute . . . . .	843
8.836.2.5 month . . . . .	843
8.836.2.6 second . . . . .	843
8.836.2.7 year . . . . .	843
8.837 unpack_dms_GetActivationState_t Struct Reference . . . . .	843
8.837.1 Detailed Description . . . . .	843
8.837.2 Field Documentation . . . . .	844
8.837.2.1 state . . . . .	844
8.838 unpack_dms_GetBandCapability_t Struct Reference . . . . .	844
8.838.1 Field Documentation . . . . .	844
8.838.1.1 BandCapability . . . . .	844
8.838.1.2 Tlvresult . . . . .	844
8.839 unpack_dms_GetCrashAction_t Struct Reference . . . . .	844

8.839.1 Field Documentation . . . . .	844
8.839.1.1 DevCrashState . . . . .	844
8.839.1.2 Tlvresult . . . . .	844
8.840unpack_dms_GetCustFeature_t Struct Reference . . . . .	844
8.840.1 Field Documentation . . . . .	845
8.840.1.1 DHCPRelayEnabled . . . . .	845
8.840.1.2 DisableIMSI . . . . .	845
8.840.1.3 GpsEnable . . . . .	845
8.840.1.4 GPSLPM . . . . .	845
8.840.1.5 GPSSel . . . . .	845
8.840.1.6 IPFamSupport . . . . .	845
8.840.1.7 IsVoiceEnabled . . . . .	845
8.840.1.8 RMAutoConnect . . . . .	845
8.840.1.9 SMSSupport . . . . .	845
8.840.1.10Tlvresult . . . . .	845
8.841unpack_dms_GetCustFeaturesV2_t Struct Reference . . . . .	845
8.841.1 Detailed Description . . . . .	845
8.841.2 Field Documentation . . . . .	845
8.841.2.1 GetCustomFeatureV2 . . . . .	845
8.841.2.2 Tlvresult . . . . .	845
8.842unpack_dms_GetDeviceCap_t Struct Reference . . . . .	846
8.842.1 Field Documentation . . . . .	846
8.842.1.1 DataServiceCapability . . . . .	846
8.842.1.2 MaxRXChannelRate . . . . .	846
8.842.1.3 MaxTXChannelRate . . . . .	846
8.842.1.4 Radiofaces . . . . .	846
8.842.1.5 RadiofacesSize . . . . .	846
8.842.1.6 SimCapability . . . . .	846
8.842.1.7 Tlvresult . . . . .	846
8.843unpack_dms_GetDeviceCapabilities_t Struct Reference . . . . .	846
8.843.1 Detailed Description . . . . .	846
8.843.2 Field Documentation . . . . .	847
8.843.2.1 dataServiceCaCapability . . . . .	847
8.843.2.2 maxRxChannelRate . . . . .	847
8.843.2.3 maxTxChannelRate . . . . .	847
8.843.2.4 Radiofaces . . . . .	847
8.843.2.5 radiofacesSize . . . . .	847
8.843.2.6 simCapability . . . . .	847
8.844unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . . .	847
8.844.1 Field Documentation . . . . .	847

8.844.1.1 String . . . . .	847
8.844.1.2 stringSize . . . . .	847
8.844.1.3 Tlvresult . . . . .	847
8.845unpack_dms_GetDeviceMfr_t Struct Reference . . . . .	847
8.845.1 Field Documentation . . . . .	847
8.845.1.1 String . . . . .	847
8.845.1.2 stringSize . . . . .	847
8.845.1.3 Tlvresult . . . . .	847
8.846unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . . .	847
8.846.1 Field Documentation . . . . .	848
8.846.1.1 esnSize . . . . .	848
8.846.1.2 ESNString . . . . .	848
8.846.1.3 imeiSize . . . . .	848
8.846.1.4 IMEIString . . . . .	848
8.846.1.5 imeiSvnSize . . . . .	848
8.846.1.6 ImeiSvnString . . . . .	848
8.846.1.7 meidSize . . . . .	848
8.846.1.8 MEIDString . . . . .	848
8.846.1.9 Tlvresult . . . . .	848
8.847unpack_dms_GetFirmwareInfo_t Struct Reference . . . . .	848
8.847.1 Detailed Description . . . . .	848
8.847.2 Field Documentation . . . . .	849
8.847.2.1 appversion_str . . . . .	849
8.847.2.2 bootversion_str . . . . .	849
8.847.2.3 carrier_str . . . . .	849
8.847.2.4 cur_carr_name . . . . .	849
8.847.2.5 cur_carr_rev . . . . .	849
8.847.2.6 modelid_str . . . . .	849
8.847.2.7 packageid_str . . . . .	849
8.847.2.8 priversion_str . . . . .	849
8.847.2.9 sku_str . . . . .	849
8.847.2.10Tlvresult . . . . .	849
8.848unpack_dms_GetFirmwareRevision_t Struct Reference . . . . .	849
8.848.1 Field Documentation . . . . .	849
8.848.1.1 amssSize . . . . .	849
8.848.1.2 AMSSString . . . . .	849
8.848.1.3 PRIString . . . . .	849
8.848.1.4 Tlvresult . . . . .	849
8.849unpack_dms_GetFirmwareRevisions_t Struct Reference . . . . .	849
8.849.1 Detailed Description . . . . .	850

8.849.2 Field Documentation . . . . .	850
8.849.2.1 amssSize . . . . .	850
8.849.2.2 AMSSString . . . . .	850
8.849.2.3 bootSize . . . . .	850
8.849.2.4 BootString . . . . .	850
8.849.2.5 priSize . . . . .	850
8.849.2.6 PRIString . . . . .	850
8.849.2.7 Tlvresult . . . . .	850
8.850unpack_dms_GetFSN_t Struct Reference . . . . .	850
8.850.1 Field Documentation . . . . .	850
8.850.1.1 String . . . . .	850
8.850.1.2 Tlvresult . . . . .	850
8.851unpack_dms_GetHardwareRevision_t Struct Reference . . . . .	850
8.851.1 Detailed Description . . . . .	850
8.851.2 Field Documentation . . . . .	851
8.851.2.1 hwVer . . . . .	851
8.852unpack_dms_GetIMSI_t Struct Reference . . . . .	851
8.852.1 Field Documentation . . . . .	851
8.852.1.1 imsi . . . . .	851
8.852.1.2 Tlvresult . . . . .	851
8.853unpack_dms_GetModelID_t Struct Reference . . . . .	851
8.853.1 Detailed Description . . . . .	851
8.853.2 Field Documentation . . . . .	851
8.853.2.1 modelid . . . . .	851
8.853.2.2 Tlvresult . . . . .	851
8.854unpack_dms_GetNetworkTime_t Struct Reference . . . . .	851
8.854.1 Detailed Description . . . . .	851
8.854.2 Field Documentation . . . . .	852
8.854.2.1 source . . . . .	852
8.854.2.2 timestamp . . . . .	852
8.854.2.3 Tlvresult . . . . .	852
8.855unpack_dms_GetPower_t Struct Reference . . . . .	852
8.855.1 Detailed Description . . . . .	852
8.855.2 Field Documentation . . . . .	852
8.855.2.1 HardwareControlledMode . . . . .	852
8.855.2.2 OfflineReason . . . . .	852
8.855.2.3 OperationMode . . . . .	852
8.855.2.4 Tlvresult . . . . .	853
8.856unpack_dms_GetPRLVersion_t Struct Reference . . . . .	853
8.856.1 Field Documentation . . . . .	853

8.856.1.1 Tlvresult . . . . .	853
8.856.1.2 u16PRLVersion . . . . .	853
8.856.1.3 u8PRLPreference . . . . .	853
8.857unpack_dms_GetSerialNumbers_t Struct Reference . . . . .	853
8.857.1 Detailed Description . . . . .	853
8.857.2 Field Documentation . . . . .	853
8.857.2.1 esn . . . . .	853
8.857.2.2 imei_no . . . . .	853
8.857.2.3 imeisv_svn . . . . .	853
8.857.2.4 meid . . . . .	853
8.858unpack_dms_GetUSBComp_t Struct Reference . . . . .	853
8.858.1 Field Documentation . . . . .	854
8.858.1.1 NumSupUSBComps . . . . .	854
8.858.1.2 SupUSBComps . . . . .	854
8.858.1.3 Tlvresult . . . . .	854
8.858.1.4 USBComp . . . . .	854
8.859unpack_dms_GetVoiceNumber_t Struct Reference . . . . .	854
8.859.1 Field Documentation . . . . .	854
8.859.1.1 MIN . . . . .	854
8.859.1.2 minSize . . . . .	854
8.859.1.3 Tlvresult . . . . .	854
8.859.1.4 VoiceNumber . . . . .	854
8.859.1.5 voiceNumberSize . . . . .	854
8.860unpack_dms_SetCrashAction_t Struct Reference . . . . .	854
8.860.1 Detailed Description . . . . .	854
8.860.2 Field Documentation . . . . .	854
8.860.2.1 notused . . . . .	854
8.861unpack_dms_SetCustFeature_t Struct Reference . . . . .	854
8.861.1 Field Documentation . . . . .	855
8.861.1.1 Tlvresult . . . . .	855
8.862unpack_dms_SetCustFeaturesV2_t Struct Reference . . . . .	855
8.862.1 Detailed Description . . . . .	855
8.862.2 Field Documentation . . . . .	855
8.862.2.1 Tlvresult . . . . .	855
8.863unpack_dms_SetEventReport_ind_t Struct Reference . . . . .	855
8.863.1 Detailed Description . . . . .	855
8.863.2 Field Documentation . . . . .	856
8.863.2.1 ActivationStatusTlv . . . . .	856
8.863.2.2 OperatingModeTlv . . . . .	856
8.863.2.3 Tlvresult . . . . .	856

8.864unpack_dms_SetEventReport_t Struct Reference	856
8.864.1 Field Documentation	856
8.864.1.1 Tlvresult	856
8.865unpack_dms_SetFirmwarePreference_t Struct Reference	856
8.865.1 Field Documentation	856
8.865.1.1 Tlvresult	856
8.866unpack_dms_SetPower_t Struct Reference	856
8.866.1 Field Documentation	856
8.866.1.1 Tlvresult	856
8.867unpack_dms_SetUSBComp_t Struct Reference	856
8.867.1 Field Documentation	856
8.867.1.1 Tlvresult	856
8.868unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference	856
8.868.1 Detailed Description	857
8.868.2 Field Documentation	857
8.868.2.1 source	857
8.868.2.2 Tlvresult	857
8.868.2.3 type	857
8.869unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference	857
8.869.1 Detailed Description	858
8.869.2 Field Documentation	858
8.869.2.1 source	858
8.869.2.2 Tlvresult	858
8.869.2.3 type	858
8.870unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference	858
8.870.1 Detailed Description	858
8.870.2 Field Documentation	859
8.870.2.1 Tlvresult	859
8.871unpack_dms_SLQSGetBandCapability_t Struct Reference	859
8.871.1 Detailed Description	859
8.871.2 Field Documentation	862
8.871.2.1 bandCapability	862
8.871.2.2 is_LteBandCapability_Available	862
8.871.2.3 is_TdsBandCapability_Available	862
8.871.2.4 LteBandCapability	862
8.871.2.5 TdsBandCapability	862
8.872unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	862
8.872.1 Detailed Description	862
8.872.2 Field Documentation	862
8.872.2.1 Tlvresult	862

8.873unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	862
8.873.1 Detailed Description	863
8.873.2 Field Documentation	863
8.873.2.1 pGetDyingGaspCfg	863
8.873.2.2 Tlvresult	863
8.874unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	863
8.874.1 Detailed Description	863
8.874.2 Field Documentation	863
8.874.2.1 pGetDyingGaspStatistics	863
8.874.2.2 Tlvresult	863
8.875unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	863
8.875.1 Detailed Description	863
8.875.2 Field Documentation	864
8.875.2.1 carrier	864
8.875.2.2 fwvers	864
8.875.2.3 numEntries	864
8.875.2.4 pCurrImgInfo	864
8.875.2.5 pkgver	864
8.875.2.6 priver	864
8.876unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference	864
8.876.1 Detailed Description	864
8.876.2 Field Documentation	865
8.876.2.1 imgType	865
8.876.2.2 logString	865
8.876.2.3 refData	865
8.876.2.4 refString	865
8.876.2.5 ResCode	865
8.876.2.6 Tlvresult	866
8.877unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference	866
8.877.1 Detailed Description	866
8.877.2 Field Documentation	866
8.877.2.1 Tlvresult	866
8.878unpack_dms_UIMGetICCID_t Struct Reference	866
8.878.1 Detailed Description	866
8.878.2 Field Documentation	866
8.878.2.1 String	866
8.878.2.2 stringSize	866
8.878.2.3 Tlvresult	867
8.879unpack_fms_GetImagesPreference_t Struct Reference	867
8.879.1 Detailed Description	867

8.879.2 Field Documentation . . . . .	867
8.879.2.1 ImageListSize . . . . .	867
8.879.2.2 pImageList . . . . .	867
8.879.2.3 Tlvresult . . . . .	867
8.880unpack_fms_GetStoredImages_t Struct Reference . . . . .	867
8.880.1 Detailed Description . . . . .	867
8.880.2 Field Documentation . . . . .	868
8.880.2.1 imageList . . . . .	868
8.880.2.2 imagelistSize . . . . .	868
8.880.2.3 Tlvresult . . . . .	868
8.881unpack_fms_SetImagesPreference_t Struct Reference . . . . .	868
8.881.1 Detailed Description . . . . .	868
8.881.2 Field Documentation . . . . .	868
8.881.2.1 ImageTypes . . . . .	868
8.881.2.2 ImageTypesSize . . . . .	868
8.881.2.3 Tlvresult . . . . .	868
8.882unpack_loc_BestAvailPos_Ind_t Struct Reference . . . . .	868
8.882.1 Detailed Description . . . . .	869
8.882.2 Field Documentation . . . . .	873
8.882.2.1 pAltitudeWrtEllipsoid . . . . .	873
8.882.2.2 pAltitudeWrtMeanSeaLevel . . . . .	873
8.882.2.3 pGpsTime . . . . .	873
8.882.2.4 pHeading . . . . .	873
8.882.2.5 pHeadingUnc . . . . .	873
8.882.2.6 pHorCirConf . . . . .	873
8.882.2.7 pHorEllpConf . . . . .	873
8.882.2.8 pHorReliability . . . . .	873
8.882.2.9 pHorUncCircular . . . . .	873
8.882.2.10pHorUncEllipseOrientAzimuth . . . . .	873
8.882.2.11pHorUncEllipseSemiMajor . . . . .	873
8.882.2.12pHorUncEllipseSemiMinor . . . . .	874
8.882.2.13pLatitude . . . . .	874
8.882.2.14pLongitude . . . . .	874
8.882.2.15pMagneticDeviation . . . . .	874
8.882.2.16pPrecisionDilution . . . . .	874
8.882.2.17pSensorDataUsage . . . . .	874
8.882.2.18pSpeedHorizontal . . . . .	874
8.882.2.19pSpeedUnc . . . . .	874
8.882.2.20pSpeedVertical . . . . .	874
8.882.2.21pSpeedVerticalUnc . . . . .	874

8.882.2.22	pSvUsedforFix	874
8.882.2.23	TechnologyMask	874
8.882.2.24	pTimeSrc	874
8.882.2.25	pTimestampUtc	874
8.882.2.26	pTimeUnc	874
8.882.2.27	pVertConfidence	874
8.882.2.28	pVertReliability	874
8.882.2.29	pVertUnc	874
8.882.2.30	pXid	874
8.882.2.31	status	874
8.882.2.32	Tlvresult	874
8.883	unpack_loc_Delete_Assist_Data_t Struct Reference	874
8.883.1	Detailed Description	874
8.883.2	Field Documentation	875
8.883.2.1	Tlvresult	875
8.884	unpack_loc_EngineState_Ind_t Struct Reference	875
8.884.1	Detailed Description	875
8.884.2	Field Documentation	875
8.884.2.1	engineState	875
8.884.2.2	Tlvresult	875
8.885	unpack_loc_EventRegister_t Struct Reference	875
8.885.1	Detailed Description	875
8.885.2	Field Documentation	876
8.885.2.1	Tlvresult	876
8.886	unpack_loc_PositionRpt_Ind_t Struct Reference	876
8.886.1	Detailed Description	876
8.886.2	Field Documentation	880
8.886.2.1	pAltitudeAssumed	880
8.886.2.2	pAltitudeWrtEllipsoid	880
8.886.2.3	pAltitudeWrtMeanSeaLevel	880
8.886.2.4	pFixId	880
8.886.2.5	pGpsTime	880
8.886.2.6	pHeading	881
8.886.2.7	pHeadingUnc	881
8.886.2.8	pHorConfidence	881
8.886.2.9	pHorReliability	881
8.886.2.10	pHorUncCircular	881
8.886.2.11	pHorUncEllipseOrientAzimuth	881
8.886.2.12	pHorUncEllipseSemiMajor	881
8.886.2.13	pHorUncEllipseSemiMinor	881

8.886.2.14pLatitude . . . . .	881
8.886.2.15pLeapSeconds . . . . .	881
8.886.2.16pLongitude . . . . .	881
8.886.2.17pMagneticDeviation . . . . .	881
8.886.2.18pPrecisionDilution . . . . .	881
8.886.2.19pSensorDataUsage . . . . .	881
8.886.2.20pSpeedHorizontal . . . . .	881
8.886.2.21pSpeedUnc . . . . .	881
8.886.2.22pSpeedVertical . . . . .	881
8.886.2.23pSvUsedforFix . . . . .	881
8.886.2.24pTechnologyMask . . . . .	881
8.886.2.25pTimeSrc . . . . .	881
8.886.2.26pTimestampUtc . . . . .	881
8.886.2.27pTimeUnc . . . . .	881
8.886.2.28pVertConfidence . . . . .	881
8.886.2.29pVertReliability . . . . .	881
8.886.2.30pVertUnc . . . . .	881
8.886.2.31sessionId . . . . .	881
8.886.2.32sessionStatus . . . . .	881
8.886.2.33Tlvresult . . . . .	881
8.887unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . . .	882
8.887.1 Detailed Description . . . . .	882
8.887.2 Field Documentation . . . . .	882
8.887.2.1 status . . . . .	882
8.887.2.2 Tlvresult . . . . .	882
8.888unpack_loc_SetExtPowerState_t Struct Reference . . . . .	882
8.888.1 Detailed Description . . . . .	882
8.888.2 Field Documentation . . . . .	883
8.888.2.1 Tlvresult . . . . .	883
8.889unpack_loc_SetOperationMode_t Struct Reference . . . . .	883
8.889.1 Detailed Description . . . . .	883
8.889.2 Field Documentation . . . . .	883
8.889.2.1 Tlvresult . . . . .	883
8.890unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . . .	883
8.890.1 Detailed Description . . . . .	883
8.890.2 Field Documentation . . . . .	883
8.890.2.1 Tlvresult . . . . .	883
8.891unpack_loc_Start_t Struct Reference . . . . .	883
8.891.1 Detailed Description . . . . .	884
8.891.2 Field Documentation . . . . .	884

8.891.2.1 Tlvresult . . . . .	884
8.892unpack_loc_Stop_t Struct Reference . . . . .	884
8.892.1 Detailed Description . . . . .	884
8.892.2 Field Documentation . . . . .	884
8.892.2.1 Tlvresult . . . . .	884
8.893unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . . .	884
8.893.1 Detailed Description . . . . .	884
8.893.2 Field Documentation . . . . .	885
8.893.2.1 Application . . . . .	885
8.893.2.2 Broadcast . . . . .	885
8.893.2.3 CustomSCP . . . . .	885
8.893.2.4 ForceRev0 . . . . .	885
8.893.2.5 Protocol . . . . .	885
8.893.2.6 RegForeignNID . . . . .	885
8.893.2.7 RegForeignSID . . . . .	885
8.893.2.8 RegHomeSID . . . . .	885
8.893.2.9 Roaming . . . . .	885
8.893.2.10SCI . . . . .	885
8.893.2.11SCM . . . . .	885
8.894unpack_nas_GetHomeNetwork_t Struct Reference . . . . .	885
8.894.1 Detailed Description . . . . .	885
8.894.2 Field Documentation . . . . .	886
8.894.2.1 mcc . . . . .	886
8.894.2.2 mnc . . . . .	886
8.894.2.3 name . . . . .	886
8.894.2.4 nid . . . . .	886
8.894.2.5 sid . . . . .	886
8.895unpack_nas_GetNetworkPreference_t Struct Reference . . . . .	886
8.895.1 Detailed Description . . . . .	886
8.895.2 Field Documentation . . . . .	887
8.895.2.1 ActiveTechPref . . . . .	887
8.895.2.2 Duration . . . . .	887
8.895.2.3 PersistentTechPref . . . . .	887
8.895.2.4 Tlvresult . . . . .	887
8.896unpack_nas_GetRFInfo_t Struct Reference . . . . .	887
8.896.1 Detailed Description . . . . .	887
8.896.2 Field Documentation . . . . .	887
8.896.2.1 instancesSize . . . . .	887
8.896.2.2 RFBandInfoElements . . . . .	887
8.897unpack_nas_GetServingNetwork_t Struct Reference . . . . .	887

8.897.1 Detailed Description	888
8.897.2 Field Documentation	888
8.897.2.1 CSDomain	888
8.897.2.2 DataCaps	888
8.897.2.3 DataCapsLen	888
8.897.2.4 MCC	888
8.897.2.5 MNC	888
8.897.2.6 Name	888
8.897.2.7 nameSize	888
8.897.2.8 PSDomain	888
8.897.2.9 Radiolfaces	888
8.897.2.10RadiolfacesSize	888
8.897.2.11RAN	888
8.897.2.12RegistrationState	888
8.897.2.13Roaming	888
8.898unpack_nas_GetServingNetworkCapabilities_t Struct Reference	888
8.898.1 Detailed Description	889
8.898.2 Field Documentation	889
8.898.2.1 DataCaps	889
8.898.2.2 DataCapsLen	889
8.899unpack_nas_GetSignalStrengths_t Struct Reference	889
8.899.1 Detailed Description	889
8.899.2 Field Documentation	889
8.899.2.1 len	889
8.899.2.2 radio	889
8.899.2.3 rssi	889
8.900unpack_nas_PerformNetworkScan_t Struct Reference	889
8.900.1 Detailed Description	889
8.900.2 Field Documentation	890
8.900.2.1 p3GppNetworkInfoInstances	890
8.900.2.2 p3GppNetworkInstanceSize	890
8.900.2.3 pPCSInstance	890
8.900.2.4 pPCSInstanceSize	890
8.900.2.5 pRATInstance	890
8.900.2.6 pRATInstanceSize	890
8.900.2.7 pScanResult	890
8.901unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	890
8.901.1 Detailed Description	890
8.901.2 Field Documentation	890
8.901.2.1 dataCaps	890

8.901.2.2 dataCapsSize . . . . .	890
8.902unpack_nas_SetEventReportInd_t Struct Reference . . . . .	890
8.902.1 Detailed Description . . . . .	890
8.902.2 Field Documentation . . . . .	891
8.902.2.1 RFTlv . . . . .	891
8.902.2.2 RRTlv . . . . .	891
8.902.2.3 SLQSSSTlv . . . . .	891
8.902.2.4 SSTlv . . . . .	891
8.903unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference . . . . .	891
8.903.1 Detailed Description . . . . .	891
8.903.2 Field Documentation . . . . .	891
8.903.2.1 sPhyCaAggPcellInfo . . . . .	891
8.903.2.2 sPhyCaAggScellIDBw . . . . .	892
8.903.2.3 sPhyCaAggScellIndex . . . . .	892
8.903.2.4 sPhyCaAggScellIndType . . . . .	892
8.903.2.5 sPhyCaAggScellInfo . . . . .	892
8.904unpack_nas_SetNetworkPreference_t Struct Reference . . . . .	892
8.904.1 Detailed Description . . . . .	892
8.904.2 Field Documentation . . . . .	892
8.904.2.1 Tlvresult . . . . .	892
8.905unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . . .	893
8.905.1 Detailed Description . . . . .	893
8.905.2 Field Documentation . . . . .	893
8.905.2.1 roaming . . . . .	893
8.906unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . . .	893
8.906.1 Detailed Description . . . . .	893
8.906.2 Field Documentation . . . . .	893
8.906.2.1 SSInfo . . . . .	893
8.906.2.2 Tlvresult . . . . .	893
8.907unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . . .	893
8.907.1 Detailed Description . . . . .	894
8.907.2 Field Documentation . . . . .	894
8.907.2.1 LTECphyCAInfo . . . . .	894
8.907.2.2 Tlvresult . . . . .	894
8.908unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . . .	894
8.908.1 Detailed Description . . . . .	894
8.908.2 Field Documentation . . . . .	894
8.908.2.1 p3GPP2TimeInfo . . . . .	894
8.908.2.2 p3GPPTimeInfo . . . . .	894
8.909unpack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	894

8.909.1 Field Documentation	895
8.909.1.1 longName	895
8.909.1.2 longNameCI	895
8.909.1.3 longNameEn	895
8.909.1.4 longNameLen	895
8.909.1.5 longNameSB	895
8.909.1.6 shortName	895
8.909.1.7 shortNameCI	895
8.909.1.8 shortNameEn	895
8.909.1.9 shortNameLen	895
8.909.1.10shortNameSB	895
8.909.1.11spn	895
8.909.1.12spnEncoding	895
8.909.1.13spnLength	895
8.910unpack_nas_SLQSGetservingSystem_t Struct Reference	895
8.910.1 Detailed Description	896
8.910.2 Field Documentation	896
8.910.2.1 BasestationID	897
8.910.2.2 BasestationLatitude	897
8.910.2.3 BasestationLongitude	897
8.910.2.4 CallBarStatus	897
8.910.2.5 CDMA_P_Rev	897
8.910.2.6 CDMASystemInfoExt	897
8.910.2.7 CellID	897
8.910.2.8 ConcSvcInfo	897
8.910.2.9 CurrentPLMN	897
8.910.2.10DataSrvCapabilities	897
8.910.2.11DefaultRoamInd	897
8.910.2.12DetailedSvcInfo	897
8.910.2.13DTMInd	897
8.910.2.14Gpp2TimeZone	897
8.910.2.15GppNetworkDSTAdjustment	897
8.910.2.16GppTimeZone	897
8.910.2.17HdrPersonality	897
8.910.2.18Lac	897
8.910.2.19NetworkID	897
8.910.2.20PRLInd	897
8.910.2.21RoamIndicatorVal	897
8.910.2.22RoamingIndicatorList	897
8.910.2.23ServingSystem	897

8.910.2.24	SystemID	897
8.910.2.25	TrackAreaCode	897
8.911	unpack_nas_SLQSGetSignalStrength_t Struct Reference	897
8.911.1	Detailed Description	898
8.911.2	Field Documentation	898
8.911.2.1	ecioList	898
8.911.2.2	ecioListLen	898
8.911.2.3	errorRateList	898
8.911.2.4	errorRateListLen	898
8.911.2.5	lo	898
8.911.2.6	ltsrps	898
8.911.2.7	ltsnr	898
8.911.2.8	rsrqInfo	898
8.911.2.9	rxSignalStrengthList	898
8.911.2.10	rxSignalStrengthListLen	898
8.911.2.11	signalStrengthReqMask	898
8.911.2.12	sinr	898
8.912	unpack_nas_SLQSGetSysInfo_t Struct Reference	899
8.912.1	Detailed Description	899
8.912.2	Field Documentation	901
8.912.2.1	pAddCDMASysInfo	901
8.912.2.2	pAddGSMSysInfo	901
8.912.2.3	pAddHDRSysInfo	901
8.912.2.4	pAddLTESysInfo	901
8.912.2.5	pAddWCDMASysInfo	901
8.912.2.6	pCDMASrvStatusInfo	901
8.912.2.7	pCDMASysInfo	901
8.912.2.8	pGSMCallBarringSysInfo	901
8.912.2.9	pGSMCipherDomainSysInfo	901
8.912.2.10	pGSMSrvStatusInfo	901
8.912.2.11	pGSMSysInfo	901
8.912.2.12	pHDRSrvStatusInfo	901
8.912.2.13	pHDRSysInfo	901
8.912.2.14	pLTESrvStatusInfo	901
8.912.2.15	pLTESysInfo	901
8.912.2.16	pLTEVoiceSupportSysInfo	901
8.912.2.17	pWCDMACallBarringSysInfo	901
8.912.2.18	pWCDMACipherDomainSysInfo	901
8.912.2.19	pWCDMASrvStatusInfo	901
8.912.2.20	pWCDMASysInfo	901

8.913unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . . .	901
8.913.1 Detailed Description . . . . .	902
8.913.2 Field Documentation . . . . .	905
8.913.2.1 pBandPref . . . . .	905
8.913.2.2 pEmerMode . . . . .	905
8.913.2.3 pGWAcqOrderPref . . . . .	905
8.913.2.4 pLTEBandPref . . . . .	905
8.913.2.5 pModePref . . . . .	905
8.913.2.6 pNetSelPref . . . . .	905
8.913.2.7 pPRLPref . . . . .	905
8.913.2.8 pRoamPref . . . . .	905
8.913.2.9 pSrvDomainPref . . . . .	905
8.914unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . . .	905
8.914.1 Detailed Description . . . . .	906
8.914.2 Field Documentation . . . . .	906
8.914.2.1 pCDMAInfo . . . . .	906
8.914.2.2 pGERANInfo . . . . .	906
8.914.2.3 pLTEInfoInterfreq . . . . .	906
8.914.2.4 pLTEInfoIntrafreq . . . . .	906
8.914.2.5 pLTEInfoNeighboringGSM . . . . .	906
8.914.2.6 pLTEInfoNeighboringWCDMA . . . . .	906
8.914.2.7 pUMTSCellID . . . . .	907
8.914.2.8 pUMTSInfo . . . . .	907
8.914.2.9 pWCDMAInfoLTENeighborCell . . . . .	907
8.915unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . . .	907
8.915.1 Detailed Description . . . . .	907
8.915.2 Field Documentation . . . . .	907
8.915.2.1 CDMASSInfo . . . . .	907
8.915.2.2 GSMSSInfo . . . . .	907
8.915.2.3 HDRSSInfo . . . . .	907
8.915.2.4 LTESInfo . . . . .	907
8.915.2.5 WCDMASSInfo . . . . .	907
8.916unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . . .	907
8.916.1 Detailed Description . . . . .	907
8.916.2 Field Documentation . . . . .	908
8.916.2.1 pDayltSavAdj . . . . .	908
8.916.2.2 pRadioInterface . . . . .	908
8.916.2.3 pTimeZone . . . . .	908
8.916.2.4 universalTime . . . . .	908
8.917unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . . .	908

8.917.1 Detailed Description . . . . .	908
8.917.2 Field Documentation . . . . .	909
8.917.2.1 pCDMASigInfo . . . . .	909
8.917.2.2 pGSMSigInfo . . . . .	909
8.917.2.3 pHDRSigInfo . . . . .	909
8.917.2.4 pLTESigInfo . . . . .	909
8.917.2.5 pRscp . . . . .	909
8.917.2.6 pTDSCDMASigInfoExt . . . . .	909
8.917.2.7 pWCDMASigInfo . . . . .	909
8.918unpack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . . .	909
8.918.1 Detailed Description . . . . .	909
8.918.2 Field Documentation . . . . .	909
8.918.2.1 commonInfo . . . . .	909
8.918.2.2 pLTEInfo . . . . .	909
8.919unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . . .	910
8.919.1 Detailed Description . . . . .	910
8.919.2 Field Documentation . . . . .	910
8.919.2.1 Info . . . . .	910
8.919.2.2 Tlvresult . . . . .	910
8.920unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . . .	910
8.920.1 Detailed Description . . . . .	910
8.920.2 Field Documentation . . . . .	910
8.920.2.1 Info . . . . .	910
8.920.2.2 Tlvresult . . . . .	910
8.921unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . . .	910
8.921.1 Detailed Description . . . . .	911
8.921.2 Field Documentation . . . . .	911
8.921.2.1 CQIValueCW0 . . . . .	911
8.921.2.2 CQIValueCW1 . . . . .	911
8.921.2.3 ValidityCW0 . . . . .	911
8.921.2.4 ValidityCW1 . . . . .	911
8.922unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . . .	911
8.922.1 Detailed Description . . . . .	912
8.922.2 Field Documentation . . . . .	913
8.922.2.1 pAddCDMASysInfo . . . . .	913
8.922.2.2 pAddGSMSysInfo . . . . .	913
8.922.2.3 pAddHDRSysInfo . . . . .	913
8.922.2.4 pAddLTESysInfo . . . . .	913
8.922.2.5 pAddWCDMASysInfo . . . . .	914
8.922.2.6 pCDMASrvStatusInfo . . . . .	914

8.922.2.7 pCDMASysInfo . . . . .	914
8.922.2.8 pGSMCallBarringSysInfo . . . . .	914
8.922.2.9 pGSMCipherDomainSysInfo . . . . .	914
8.922.2.10pGSMSrvStatusInfo . . . . .	914
8.922.2.11pGSMSysInfo . . . . .	914
8.922.2.12pHDRSrvStatusInfo . . . . .	914
8.922.2.13pHDRSysInfo . . . . .	914
8.922.2.14pLTERsrvStatusInfo . . . . .	914
8.922.2.15pLTERSysInfo . . . . .	914
8.922.2.16pLTEVoiceSupportSysInfo . . . . .	914
8.922.2.17pSysInfoNoChange . . . . .	914
8.922.2.18pWCDMACallBarringSysInfo . . . . .	914
8.922.2.19pWCDMACipherDomainSysInfo . . . . .	914
8.922.2.20pWCDMASrvStatusInfo . . . . .	914
8.922.2.21pWCDMASysInfo . . . . .	914
8.923unpack_omaDmConfigTlv_t Struct Reference . . . . .	914
8.923.1 Detailed Description . . . . .	914
8.923.2 Field Documentation . . . . .	915
8.923.2.1 alertmsg . . . . .	915
8.923.2.2 alertmsglength . . . . .	915
8.923.2.3 state . . . . .	915
8.923.2.4 userInputReq . . . . .	915
8.923.2.5 userInputTimeout . . . . .	915
8.924unpack_omaDmFotaTlv_t Struct Reference . . . . .	915
8.924.1 Detailed Description . . . . .	915
8.924.2 Field Documentation . . . . .	917
8.924.2.1 description . . . . .	917
8.924.2.2 descriptionlength . . . . .	917
8.924.2.3 fwdloadsize . . . . .	917
8.924.2.4 fwloadComplete . . . . .	917
8.924.2.5 namelength . . . . .	917
8.924.2.6 package_name . . . . .	917
8.924.2.7 sessionType . . . . .	917
8.924.2.8 severity . . . . .	917
8.924.2.9 state . . . . .	917
8.924.2.10updateCompleteStatus . . . . .	917
8.924.2.11userInputReq . . . . .	917
8.924.2.12userInputTimeout . . . . .	917
8.924.2.13version . . . . .	917
8.924.2.14versionlength . . . . .	917

8.925unpack_omaDmNotificationsTlv_t Struct Reference	917
8.925.1 Field Documentation	917
8.925.1.1 notification	917
8.925.1.2 sessionStatus	917
8.926unpack_qmi_t Struct Reference	917
8.926.1 Detailed Description	918
8.926.2 Field Documentation	918
8.926.2.1 msgid	918
8.926.2.2 type	918
8.926.2.3 xid	918
8.927unpack_qos_dataRate_t Struct Reference	918
8.927.1 Detailed Description	918
8.927.2 Field Documentation	918
8.927.2.1 dataRateMax	918
8.927.2.2 guaranteedRate	918
8.928unpack_qos_IPv4Addr_t Struct Reference	918
8.928.1 Detailed Description	918
8.928.2 Field Documentation	919
8.928.2.1 addr	919
8.928.2.2 subnetMask	919
8.929unpack_qos_IPv6Addr_t Struct Reference	919
8.929.1 Detailed Description	919
8.929.2 Field Documentation	919
8.929.2.1 addr	919
8.929.2.2 prefixLen	919
8.930unpack_qos_IPv6TrafCls_t Struct Reference	919
8.930.1 Detailed Description	919
8.930.2 Field Documentation	920
8.930.2.1 mask	920
8.930.2.2 val	920
8.931unpack_qos_pktErrRate_t Struct Reference	920
8.931.1 Detailed Description	920
8.931.2 Field Documentation	920
8.931.2.1 exponent	920
8.931.2.2 multiplier	920
8.932unpack_qos_Port_t Struct Reference	920
8.932.1 Detailed Description	920
8.932.2 Field Documentation	921
8.932.2.1 port	921
8.932.2.2 range	921

8.933unpack_qos_QosFlowInfo_t Struct Reference . . . . .	921
8.933.1 Detailed Description . . . . .	921
8.933.2 Field Documentation . . . . .	922
8.933.2.1 BearerID . . . . .	922
8.933.2.2 is_RxQFlowGranted_Available . . . . .	922
8.933.2.3 is_TxQFlowGranted_Available . . . . .	922
8.933.2.4 NumRxFilters . . . . .	922
8.933.2.5 NumTxFilters . . . . .	922
8.933.2.6 QFlowState . . . . .	922
8.933.2.7 RxQFilter . . . . .	922
8.933.2.8 RxQFlowGranted . . . . .	922
8.933.2.9 TxQFilter . . . . .	922
8.933.2.10TxQFlowGranted . . . . .	922
8.934unpack_qos_QosFlowInfoState_t Struct Reference . . . . .	922
8.934.1 Detailed Description . . . . .	922
8.934.2 Field Documentation . . . . .	923
8.934.2.1 id . . . . .	923
8.934.2.2 isNewFlow . . . . .	923
8.934.2.3 state . . . . .	923
8.935unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . . .	923
8.935.1 Detailed Description . . . . .	923
8.935.2 Field Documentation . . . . .	923
8.935.2.1 NWQoSStatus . . . . .	923
8.936unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . . .	923
8.936.1 Detailed Description . . . . .	924
8.936.2 Field Documentation . . . . .	924
8.936.2.1 ambr_dl . . . . .	924
8.936.2.2 ambr_dl_ext . . . . .	924
8.936.2.3 ambr_dl_ext2 . . . . .	924
8.936.2.4 ambr_ul . . . . .	924
8.936.2.5 ambr_ul_ext . . . . .	924
8.936.2.6 ambr_ul_ext2 . . . . .	924
8.936.2.7 apnId . . . . .	925
8.937unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . . .	925
8.937.1 Detailed Description . . . . .	925
8.937.2 Field Documentation . . . . .	926
8.937.2.1 apnId . . . . .	926
8.937.2.2 numQosFlow . . . . .	926
8.937.2.3 qosFlow . . . . .	926
8.937.2.4 total_rx_bytes . . . . .	926

8.937.2.5 total_rx_pkt . . . . .	926
8.937.2.6 total_tx_bytes . . . . .	926
8.937.2.7 total_tx_bytes_drp . . . . .	926
8.937.2.8 total_tx_pkt . . . . .	926
8.937.2.9 total_tx_pkt_drp . . . . .	926
8.938unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . . .	926
8.938.1 Detailed Description . . . . .	926
8.938.2 Field Documentation . . . . .	926
8.938.2.1 NumFlows . . . . .	926
8.938.2.2 QosFlowInfo . . . . .	926
8.939unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . . .	926
8.939.1 Detailed Description . . . . .	927
8.939.2 Field Documentation . . . . .	927
8.939.2.1 status . . . . .	927
8.940unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . . .	927
8.940.1 Detailed Description . . . . .	927
8.940.2 Field Documentation . . . . .	927
8.940.2.1 event . . . . .	927
8.941unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . . .	927
8.941.1 Detailed Description . . . . .	927
8.941.2 Field Documentation . . . . .	928
8.941.2.1 event . . . . .	928
8.941.2.2 id . . . . .	928
8.941.2.3 reason . . . . .	928
8.941.2.4 status . . . . .	928
8.942unpack_qos_swiQosFilter_t Struct Reference . . . . .	929
8.942.1 Detailed Description . . . . .	929
8.942.2 Field Documentation . . . . .	931
8.942.2.1 EspSpi . . . . .	931
8.942.2.2 Id . . . . .	931
8.942.2.3 index . . . . .	931
8.942.2.4 IPv4DstAddr . . . . .	931
8.942.2.5 IPv4SrcAddr . . . . .	931
8.942.2.6 IPv4Tos . . . . .	931
8.942.2.7 IPv6DstAddr . . . . .	931
8.942.2.8 IPv6Label . . . . .	931
8.942.2.9 IPv6SrcAddr . . . . .	931
8.942.2.10IPv6TrafCls . . . . .	931
8.942.2.11is_EspSpi_Available . . . . .	931
8.942.2.12s_Id_Available . . . . .	931

8.942.2.13s_IPv4DstAddr_Available . . . . .	931
8.942.2.14s_IPv4SrcAddr_Available . . . . .	931
8.942.2.15s_IPv4Tos_Available . . . . .	931
8.942.2.16s_IPv6DstAddr_Available . . . . .	931
8.942.2.17s_IPv6Label_Available . . . . .	931
8.942.2.18s_IPv6SrcAddr_Available . . . . .	931
8.942.2.19s_IPv6TrafCls_Available . . . . .	931
8.942.2.20s_NxtHdrProto_Available . . . . .	931
8.942.2.21s_Precedence_Available . . . . .	931
8.942.2.22s_TCPDstPort_Available . . . . .	931
8.942.2.23s_TCPSrcPort_Available . . . . .	932
8.942.2.24s_TranDstPort_Available . . . . .	932
8.942.2.25s_TranSrcPort_Available . . . . .	932
8.942.2.26s_UDPDstPort_Available . . . . .	932
8.942.2.27s_UDPSrcPort_Available . . . . .	932
8.942.2.28s_NxtHdrProto . . . . .	932
8.942.2.29s_Precedence . . . . .	932
8.942.2.30s_TCPDstPort . . . . .	932
8.942.2.31s_TCPSrcPort . . . . .	932
8.942.2.32s_TranDstPort . . . . .	932
8.942.2.33s_TranSrcPort . . . . .	932
8.942.2.34s_UDPDstPort . . . . .	932
8.942.2.35s_UDPSrcPort . . . . .	932
8.942.2.36s_version . . . . .	932
8.943unpack_qos_swiQosFlow_t Struct Reference . . . . .	932
8.943.1 Detailed Description . . . . .	933
8.943.2 Field Documentation . . . . .	935
8.943.2.1 DataRate . . . . .	935
8.943.2.2 index . . . . .	935
8.943.2.3 is_DataRate_Available . . . . .	935
8.943.2.4 is_Jitter_Available . . . . .	935
8.943.2.5 is_Latency_Available . . . . .	935
8.943.2.6 is_LteQci_Available . . . . .	935
8.943.2.7 is_MaxAllowedPktSz_Available . . . . .	935
8.943.2.8 is_MinPolicedPktSz_Available . . . . .	935
8.943.2.9 is_PktErrRate_Available . . . . .	935
8.943.2.10s_ProfileId3GPP2_Available . . . . .	935
8.943.2.11s_TokenBucket_Available . . . . .	935
8.943.2.12s_TrafficClass_Available . . . . .	935
8.943.2.13s_val_3GPP2Pri_Available . . . . .	935

8.943.2.14s_val_3GPPImCn_Available . . . . .	935
8.943.2.15s_val_3GPPResResidualBER_Available . . . . .	935
8.943.2.16s_val_3GPPSigInd_Available . . . . .	935
8.943.2.17s_val_3GPPTraHdlPri_Available . . . . .	935
8.943.2.18Jitter . . . . .	935
8.943.2.19Latency . . . . .	936
8.943.2.20LteQci . . . . .	936
8.943.2.21MaxAllowedPktSz . . . . .	936
8.943.2.22MinPolicedPktSz . . . . .	936
8.943.2.23PktErrRate . . . . .	936
8.943.2.24ProfileId3GPP2 . . . . .	936
8.943.2.25TokenBucket . . . . .	936
8.943.2.26TrafficClass . . . . .	936
8.943.2.27val_3GPP2Pri . . . . .	936
8.943.2.28val_3GPPImCn . . . . .	936
8.943.2.29val_3GPPResResidualBER . . . . .	936
8.943.2.30val_3GPPSigInd . . . . .	936
8.943.2.31val_3GPPTraHdlPri . . . . .	936
8.944unpack_qos_tokenBucket_t Struct Reference . . . . .	936
8.944.1 Detailed Description . . . . .	936
8.944.2 Field Documentation . . . . .	936
8.944.2.1 bucketSz . . . . .	936
8.944.2.2 peakRate . . . . .	936
8.944.2.3 tokenRate . . . . .	936
8.945unpack_qos_Tos_t Struct Reference . . . . .	937
8.945.1 Detailed Description . . . . .	937
8.945.2 Field Documentation . . . . .	937
8.945.2.1 mask . . . . .	937
8.945.2.2 val . . . . .	937
8.946unpack_QosFlowStat_t Struct Reference . . . . .	937
8.946.1 Detailed Description . . . . .	937
8.946.2 Field Documentation . . . . .	938
8.946.2.1 bearerId . . . . .	938
8.946.2.2 tx_bytes . . . . .	938
8.946.2.3 tx_bytes_drp . . . . .	938
8.946.2.4 tx_pkt . . . . .	938
8.946.2.5 tx_pkt_drp . . . . .	938
8.947unpack_sms_SendSMS_t Struct Reference . . . . .	938
8.947.1 Detailed Description . . . . .	938
8.947.2 Field Documentation . . . . .	938

8.947.2.1 messageFailureCode . . . . .	938
8.947.2.2 messageID . . . . .	938
8.948unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . . .	938
8.948.1 Detailed Description . . . . .	939
8.948.2 Field Documentation . . . . .	939
8.948.2.1 ETWSPLMNTlv . . . . .	939
8.948.2.2 ETWSTlv . . . . .	939
8.948.2.3 IMSTlv . . . . .	939
8.948.2.4 MMTlv . . . . .	939
8.948.2.5 NewMMTlv . . . . .	939
8.948.2.6 SMSCTlv . . . . .	939
8.948.2.7 TRMessageTlv . . . . .	939
8.949unpack_sms_SetNewSMSCallback_t Struct Reference . . . . .	939
8.950unpack_sms_SLQSDeleteSMS_t Struct Reference . . . . .	940
8.951unpack_sms_SLQSGetSMS_t Struct Reference . . . . .	940
8.951.1 Detailed Description . . . . .	940
8.951.2 Field Documentation . . . . .	940
8.951.2.1 message . . . . .	940
8.951.2.2 messageFormat . . . . .	940
8.951.2.3 messageSize . . . . .	940
8.951.2.4 messageTag . . . . .	940
8.952unpack_sms_SLQSGetSMSList_t Struct Reference . . . . .	940
8.952.1 Detailed Description . . . . .	941
8.952.2 Field Documentation . . . . .	941
8.952.2.1 messageList . . . . .	941
8.952.2.2 messageListSize . . . . .	941
8.953unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . . .	941
8.954unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . . .	941
8.954.1 Detailed Description . . . . .	941
8.954.2 Field Documentation . . . . .	941
8.954.2.1 messageMode . . . . .	941
8.954.2.2 storageType . . . . .	942
8.955unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . . .	942
8.955.1 Detailed Description . . . . .	942
8.955.2 Field Documentation . . . . .	943
8.955.2.1 fix_rate . . . . .	943
8.955.2.2 fix_rate_reported . . . . .	943
8.955.2.3 fix_type . . . . .	943
8.955.2.4 fix_type_reported . . . . .	943
8.955.2.5 function . . . . .	943

8.955.2.6 function_reported . . . . .	943
8.955.2.7 max_dist . . . . .	943
8.955.2.8 max_dist_reported . . . . .	943
8.955.2.9 max_time . . . . .	943
8.955.2.10max_time_reported . . . . .	943
8.956unpack_swioma_SLQSOMADMAAlertCallback_ind_t Struct Reference . . . . .	943
8.956.1 Detailed Description . . . . .	943
8.956.2 Field Documentation . . . . .	944
8.956.2.1 eventType . . . . .	944
8.956.2.2 SessionInfoConfig . . . . .	944
8.956.2.3 SessionInfoFota . . . . .	944
8.956.2.4 SessionInfoNotification . . . . .	944
8.957unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	944
8.957.1 Detailed Description . . . . .	944
8.957.2 Field Documentation . . . . .	946
8.957.2.1 Date . . . . .	946
8.957.2.2 DateLength . . . . .	946
8.957.2.3 PkgDescLength . . . . .	946
8.957.2.4 PkgDescription . . . . .	946
8.957.2.5 PkgName . . . . .	946
8.957.2.6 PkgNameLength . . . . .	946
8.957.2.7 RetryCount . . . . .	946
8.957.2.8 SessionState . . . . .	946
8.957.2.9 SessionType . . . . .	946
8.957.2.10Severity . . . . .	946
8.957.2.11Source . . . . .	946
8.957.2.12SourceLength . . . . .	946
8.957.2.13Status . . . . .	947
8.957.2.14Time . . . . .	947
8.957.2.15TimeLength . . . . .	947
8.957.2.16UpdateCompleteStatus . . . . .	947
8.958unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . . .	947
8.958.1 Detailed Description . . . . .	947
8.958.2 Field Documentation . . . . .	948
8.958.2.1 Autosdm . . . . .	948
8.958.2.2 FOTAdownload . . . . .	948
8.958.2.3 FOTAUpdate . . . . .	948
8.958.2.4 FwAutoCheck . . . . .	948
8.958.2.5 OMADMEEnabled . . . . .	948
8.959unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . . .	948

8.959.1 Detailed Description . . . . .	948
8.959.2 Field Documentation . . . . .	948
8.959.2.1 FwAvailability . . . . .	949
8.960unpack_uim_ChangePin_t Struct Reference . . . . .	949
8.960.1 Detailed Description . . . . .	949
8.960.2 Field Documentation . . . . .	949
8.960.2.1 pEncryptedPIN1 . . . . .	949
8.960.2.2 pIndicationToken . . . . .	949
8.960.2.3 pRemainingRetries . . . . .	949
8.960.2.4 Tlvresult . . . . .	949
8.961unpack_uim_GetCardStatus_t Struct Reference . . . . .	949
8.961.1 Detailed Description . . . . .	950
8.961.2 Field Documentation . . . . .	950
8.961.2.1 pCardStatus . . . . .	950
8.961.2.2 pHotSwapStatus . . . . .	950
8.961.2.3 Tlvresult . . . . .	950
8.962unpack_uim_ReadTransparent_t Struct Reference . . . . .	950
8.962.1 Detailed Description . . . . .	950
8.962.2 Field Documentation . . . . .	951
8.962.2.1 pCardResult . . . . .	951
8.962.2.2 pEncryptedData . . . . .	951
8.962.2.3 pIndicationToken . . . . .	951
8.962.2.4 pReadResult . . . . .	951
8.962.2.5 Tlvresult . . . . .	951
8.963unpack_uim_SetPinProtection_t Struct Reference . . . . .	951
8.963.1 Detailed Description . . . . .	951
8.963.2 Field Documentation . . . . .	951
8.963.2.1 pEncryptedPIN1 . . . . .	951
8.963.2.2 pIndicationToken . . . . .	951
8.963.2.3 pRemainingRetries . . . . .	951
8.963.2.4 Tlvresult . . . . .	952
8.964unpack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference . . . . .	952
8.964.1 Detailed Description . . . . .	952
8.964.2 Field Documentation . . . . .	952
8.964.2.1 bNumberOfPhySlots . . . . .	952
8.964.2.2 slotsstatusChange . . . . .	952
8.965unpack_uim_SLQSUIEventRegister_t Struct Reference . . . . .	952
8.965.1 Detailed Description . . . . .	952
8.965.2 Field Documentation . . . . .	952
8.965.2.1 eventMask . . . . .	952

8.966unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference . . . . .	952
8.966.1 Detailed Description . . . . .	953
8.966.2 Field Documentation . . . . .	953
8.966.2.1 pNumberOfPhySlot . . . . .	953
8.966.2.2 pUimSlotsStatus . . . . .	953
8.967unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . . .	953
8.967.1 Detailed Description . . . . .	953
8.967.2 Field Documentation . . . . .	953
8.967.2.1 pCardStatus . . . . .	953
8.968unpack_uim_UnblockPin_t Struct Reference . . . . .	953
8.968.1 Detailed Description . . . . .	954
8.968.2 Field Documentation . . . . .	954
8.968.2.1 pEncryptedPIN1 . . . . .	954
8.968.2.2 pIndicationToken . . . . .	954
8.968.2.3 pRemainingRetries . . . . .	954
8.968.2.4 Tlvresult . . . . .	954
8.969unpack_uim_VerifyPin_t Struct Reference . . . . .	954
8.969.1 Detailed Description . . . . .	954
8.969.2 Field Documentation . . . . .	955
8.969.2.1 pEncryptedPIN1 . . . . .	955
8.969.2.2 pIndicationToken . . . . .	955
8.969.2.3 pRemainingRetries . . . . .	955
8.969.2.4 Tlvresult . . . . .	955
8.970unpack_wds_GetConnectionRate_t Struct Reference . . . . .	955
8.970.1 Detailed Description . . . . .	955
8.970.2 Field Documentation . . . . .	955
8.970.2.1 currentChannelRXRate . . . . .	955
8.970.2.2 currentChannelTXRate . . . . .	955
8.970.2.3 maxChannelRXRate . . . . .	955
8.970.2.4 maxChannelTXRate . . . . .	955
8.971unpack_wds_GetDefaultProfile_t Struct Reference . . . . .	955
8.971.1 Detailed Description . . . . .	956
8.971.2 Field Documentation . . . . .	956
8.971.2.1 apnname . . . . .	956
8.971.2.2 apnsize . . . . .	956
8.971.2.3 auth . . . . .	956
8.971.2.4 ipaddr . . . . .	956
8.971.2.5 ipaddrv6 . . . . .	956
8.971.2.6 name . . . . .	956
8.971.2.7 namesize . . . . .	956

8.971.2.8 pdptype . . . . .	956
8.971.2.9 pridns . . . . .	956
8.971.2.10pridnsv6 . . . . .	956
8.971.2.11secdns . . . . .	956
8.971.2.12secdnsv6 . . . . .	956
8.971.2.13username . . . . .	957
8.971.2.14usersize . . . . .	957
8.972unpack_wds_GetDefaultProfileNum_t Struct Reference . . . . .	957
8.972.1 Detailed Description . . . . .	957
8.972.2 Field Documentation . . . . .	957
8.972.2.1 index . . . . .	957
8.973unpack_wds_GetDormancyState_t Struct Reference . . . . .	957
8.973.1 Detailed Description . . . . .	957
8.973.2 Field Documentation . . . . .	957
8.973.2.1 dormancyState . . . . .	957
8.974unpack_wds_GetLastMobileIPError_t Struct Reference . . . . .	957
8.974.1 Detailed Description . . . . .	957
8.974.2 Field Documentation . . . . .	957
8.974.2.1 error . . . . .	957
8.975unpack_wds_GetMobileIP_t Struct Reference . . . . .	958
8.975.1 Detailed Description . . . . .	958
8.975.2 Field Documentation . . . . .	958
8.975.2.1 mipMode . . . . .	958
8.976unpack_wds_GetMobileIPProfile_t Struct Reference . . . . .	958
8.976.1 Detailed Description . . . . .	958
8.976.2 Field Documentation . . . . .	958
8.976.2.1 AAASPI . . . . .	958
8.976.2.2 AAASState . . . . .	959
8.976.2.3 address . . . . .	959
8.976.2.4 enabled . . . . .	959
8.976.2.5 HASPI . . . . .	959
8.976.2.6 HASState . . . . .	959
8.976.2.7 NAI . . . . .	959
8.976.2.8 naiSize . . . . .	959
8.976.2.9 primaryHA . . . . .	959
8.976.2.10revTunneling . . . . .	959
8.976.2.11secondaryHA . . . . .	959
8.977unpack_wds_GetPacketStatus_t Struct Reference . . . . .	959
8.977.1 Detailed Description . . . . .	959
8.977.2 Field Documentation . . . . .	960

8.977.2.1 rXDroppedCount . . . . .	960
8.977.2.2 rXOkBytesCount . . . . .	960
8.977.2.3 rXOKBytesLastCall . . . . .	960
8.977.2.4 rXPacketErrors . . . . .	960
8.977.2.5 rXPacketOverflows . . . . .	960
8.977.2.6 rXPacketSuccesses . . . . .	960
8.977.2.7 tXDroppedCount . . . . .	960
8.977.2.8 tXOkBytesCount . . . . .	960
8.977.2.9 tXOKBytesLastCall . . . . .	960
8.977.2.10 tXPacketErrors . . . . .	960
8.977.2.11 tXPacketOverflows . . . . .	960
8.977.2.12 tXPacketSuccesses . . . . .	960
8.978unpack_wds_GetSessionDuration_t Struct Reference . . . . .	960
8.978.1 Detailed Description . . . . .	960
8.978.2 Field Documentation . . . . .	960
8.978.2.1 callDuration . . . . .	960
8.979unpack_wds_GetSessionState_t Struct Reference . . . . .	960
8.979.1 Detailed Description . . . . .	961
8.979.2 Field Documentation . . . . .	961
8.979.2.1 connectionStatus . . . . .	961
8.980unpack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	961
8.981unpack_wds_SetMobileIPProfile_t Struct Reference . . . . .	961
8.982unpack_wds_SLQSCreateProfile_t Struct Reference . . . . .	961
8.982.1 Detailed Description . . . . .	961
8.982.2 Field Documentation . . . . .	961
8.982.2.1 pCreateProfileOut . . . . .	961
8.982.2.2 pProfileID . . . . .	961
8.982.2.3 Tlvresult . . . . .	961
8.983unpack_wds_SLQSDeleteProfile_t Struct Reference . . . . .	961
8.983.1 Detailed Description . . . . .	961
8.983.2 Field Documentation . . . . .	962
8.983.2.1 extendedErrorCode . . . . .	962
8.984unpack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . . .	962
8.984.1 Detailed Description . . . . .	962
8.984.2 Field Documentation . . . . .	962
8.984.2.1 _3gppRelease . . . . .	962
8.984.2.2 defaultPDNEnabled . . . . .	962
8.984.2.3 LTEAttachProfileList . . . . .	963
8.984.2.4 LTEAttachProfileListLen . . . . .	963
8.984.2.5 profileList . . . . .	963

8.985unpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	963
8.985.1 Detailed Description . . . . .	963
8.985.2 Field Documentation . . . . .	963
8.985.2.1 currNetworkInfo . . . . .	963
8.985.2.2 networkInfoLen . . . . .	963
8.985.2.3 prefNetwork . . . . .	963
8.986unpack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . . .	963
8.986.1 Detailed Description . . . . .	963
8.986.2 Field Documentation . . . . .	963
8.986.2.1 curDataBearerTechnology . . . . .	963
8.986.2.2 dataBearerMask . . . . .	964
8.986.2.3 lastCallDataBearerTechnology . . . . .	964
8.987unpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . . .	964
8.987.1 Detailed Description . . . . .	964
8.987.2 Field Documentation . . . . .	964
8.987.2.1 callEndReason . . . . .	964
8.987.2.2 channelRate . . . . .	964
8.987.2.3 connectionStatus . . . . .	964
8.987.2.4 dataBearerTech . . . . .	964
8.987.2.5 dormancyStatus . . . . .	964
8.987.2.6 lastCallDataBearerTech . . . . .	965
8.987.2.7 lastCallRXOKBytesCnt . . . . .	965
8.987.2.8 lastCallTXOKBytesCnt . . . . .	965
8.987.2.9 mdmCallDurationActive . . . . .	965
8.987.2.10rxOKBytesCount . . . . .	965
8.987.2.11txOKBytesCount . . . . .	965
8.988unpack_wds_SLQSGetProfileSettings_t Struct Reference . . . . .	965
8.988.1 Field Documentation . . . . .	965
8.988.1.1 pProfileSettings . . . . .	965
8.988.1.2 ProfileType . . . . .	965
8.988.1.3 Tlvresult . . . . .	965
8.989unpack_wds_SLQSGetRuntimeSettings_t Struct Reference . . . . .	965
8.989.1 Detailed Description . . . . .	966
8.989.2 Field Documentation . . . . .	966
8.989.2.1 APNName . . . . .	966
8.989.2.2 Authentication . . . . .	966
8.989.2.3 DomainList . . . . .	966
8.989.2.4 GPRSGrantedQoS . . . . .	966
8.989.2.5 GWAddressV4 . . . . .	966
8.989.2.6 IMCNflag . . . . .	966

8.989.2.7 IPFamilyPreference . . . . .	966
8.989.2.8 IPv4 . . . . .	967
8.989.2.9 IPV6AddrInfo . . . . .	967
8.989.2.10 IPV6GWAddrInfo . . . . .	967
8.989.2.11 Mtu . . . . .	967
8.989.2.12 PCSCFAddrPCO . . . . .	967
8.989.2.13 PCSCFFQDNAddrList . . . . .	967
8.989.2.14 PDType . . . . .	967
8.989.2.15 PrimaryDNSV4 . . . . .	967
8.989.2.16 PrimaryDNSV6 . . . . .	967
8.989.2.17 ProfileID . . . . .	967
8.989.2.18 ProfileName . . . . .	967
8.989.2.19 SecondaryDNSV4 . . . . .	967
8.989.2.20 SecondaryDNSV6 . . . . .	967
8.989.2.21 ServerAddrList . . . . .	967
8.989.2.22 SubnetMaskV4 . . . . .	967
8.989.2.23 Technology . . . . .	967
8.989.2.24 UMTSGrantedQoS . . . . .	967
8.989.2.25 Username . . . . .	967
8.990unpack_wds_SLQSMModifyProfile_t Struct Reference . . . . .	967
8.990.1 Detailed Description . . . . .	967
8.990.2 Field Documentation . . . . .	967
8.990.2.1 pExtErrorCode . . . . .	967
8.991unpack_wds_SLQSSetIPFamilyPreference_t Struct Reference . . . . .	967
8.991.1 Detailed Description . . . . .	968
8.991.2 Field Documentation . . . . .	968
8.991.2.1 Tlvresult . . . . .	968
8.992unpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference . . . . .	968
8.992.1 Detailed Description . . . . .	968
8.992.2 Field Documentation . . . . .	968
8.992.2.1 bearerID . . . . .	968
8.992.2.2 conn_status . . . . .	968
8.992.2.3 ipFamily . . . . .	969
8.992.2.4 reconfigReqd . . . . .	969
8.992.2.5 sessionEndReason . . . . .	969
8.992.2.6 techName . . . . .	969
8.992.2.7 verboseSessnEndReason . . . . .	969
8.992.2.8 verboseSessnEndReasonType . . . . .	969
8.993unpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference . . . . .	969
8.993.1 Detailed Description . . . . .	969

8.993.2 Field Documentation	970
8.993.2.1 currDBTechAvail	970
8.993.2.2 currNWInfo	970
8.993.2.3 dataSysStatAvail	970
8.993.2.4 dBTechAvail	970
8.993.2.5 dBTechnology	970
8.993.2.6 dormancyStatAvail	970
8.993.2.7 dormancyStatus	970
8.993.2.8 mipstatAvail	970
8.993.2.9 mipStatus	970
8.993.2.10netInfoLen	970
8.993.2.11prefNetwork	970
8.993.2.12atMask	970
8.993.2.13rx_bytes	970
8.993.2.14rx_pkts	970
8.993.2.15soMask	970
8.993.2.16tx_bytes	970
8.993.2.17tx_pkts	970
8.993.2.18ferStatAvail	970
8.994unpack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	970
8.994.1 Detailed Description	970
8.994.2 Field Documentation	970
8.994.2.1 pHwConfig	971
8.994.2.2 pRequestOptionList	971
8.995unpack_wds_SLQSSStartDataSession_t Struct Reference	971
8.995.1 Detailed Description	971
8.995.2 Field Documentation	971
8.995.2.1 pFailureReason	971
8.995.2.2 psid	971
8.995.2.3 pVerboseFailReasonType	971
8.995.2.4 pVerboseFailureReason	971
8.996unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	971
8.996.1 Detailed Description	972
8.996.2 Field Documentation	972
8.996.2.1 apnName	972
8.996.2.2 bearerId	972
8.996.2.3 contextId	972
8.996.2.4 ipv4Address	972
8.996.2.5 ipv4GWAddress	972
8.996.2.6 ipv6Address	972

8.996.2.7 ipv6GWAddress . . . . .	972
8.996.2.8 prDNSIPv4Address . . . . .	972
8.996.2.9 prDNSIPv6Address . . . . .	973
8.996.2.10prPCSCFIPv4Address . . . . .	973
8.996.2.11prPCSCFIPv6Address . . . . .	973
8.996.2.12seDNSIPv4Address . . . . .	973
8.996.2.13seDNSIPv6Address . . . . .	973
8.996.2.14sePCSCFIPv4Address . . . . .	973
8.996.2.15sePCSCFIPv6Address . . . . .	973
8.997UnPackGetProfileSettingOut Struct Reference . . . . .	973
8.997.1 Field Documentation . . . . .	973
8.997.1.1 curProfile . . . . .	973
8.997.1.2 pExtErrCode . . . . .	973
8.998unpackWdsProfileParam Union Reference . . . . .	973
8.998.1 Field Documentation . . . . .	973
8.998.1.1 SlqsProfile3GPP . . . . .	973
8.998.1.2 SlqsProfile3GPP2 . . . . .	973
8.999USBCompConfig Struct Reference . . . . .	973
8.999.1 Detailed Description . . . . .	973
8.999.2 Field Documentation . . . . .	974
8.999.2.1 pUSBComp . . . . .	974
8.1000USBCompParams Struct Reference . . . . .	974
8.1000.1Detailed Description . . . . .	974
8.1000.2Field Documentation . . . . .	976
8.1000.2.1pNumSupUSBComps . . . . .	976
8.1000.2.2pSupUSBComps . . . . .	976
8.1000.2.3pUSBComp . . . . .	976
8.1000USSDNoWaitIndicationInfo Struct Reference . . . . .	976
8.1001.1Detailed Description . . . . .	976
8.1001.2Field Documentation . . . . .	977
8.1001.2.1pAlphaIdentifier . . . . .	977
8.1001.2.2pError . . . . .	977
8.1001.2.3pFailureCause . . . . .	977
8.1001.2.4pUSSDData . . . . .	977
8.1002USSDRespFNetwork Struct Reference . . . . .	977
8.1002.1Detailed Description . . . . .	977
8.1002.2Field Documentation . . . . .	977
8.1002.2.1pRespData . . . . .	977
8.1002.2.2pTypeCode . . . . .	977
8.1003ISSInfo Struct Reference . . . . .	977

8.1003.1Detailed Description . . . . .	978
8.1003.2Field Documentation . . . . .	978
8.1003.2.1ussData . . . . .	978
8.1003.2.2ussDCS . . . . .	978
8.1003.2.3ussLen . . . . .	978
8.1004USSResp Struct Reference . . . . .	978
8.1004.1Field Documentation . . . . .	978
8.1004.1.1pAlphaIDInfo . . . . .	978
8.1004.1.2pCallId . . . . .	978
8.1004.1.3pCcResultType . . . . .	978
8.1004.1.4pCCSuppsType . . . . .	978
8.1004.1.5pfailureCause . . . . .	979
8.1004.1.6pUSSDInfo . . . . .	979
8.1005UIUSInfo Struct Reference . . . . .	979
8.1005.1Detailed Description . . . . .	979
8.1005.2Field Documentation . . . . .	980
8.1005.2.1UIUSData . . . . .	980
8.1005.2.2UIUSDatalen . . . . .	980
8.1005.2.3UIUSDcs . . . . .	980
8.1005.2.4UIUStype . . . . .	980
8.1006VerifyUIMPIN Struct Reference . . . . .	980
8.1006.1Detailed Description . . . . .	980
8.1006.2Field Documentation . . . . .	980
8.1006.2.1pinID . . . . .	980
8.1006.2.2pinLen . . . . .	980
8.1006.2.3pinVal . . . . .	980
8.1007VoiceALSSelectLineInfo Struct Reference . . . . .	980
8.1007.1Detailed Description . . . . .	981
8.1007.2Field Documentation . . . . .	981
8.1007.2.1lineValue . . . . .	981
8.1008VoiceALSSetLineSwitchInfo Struct Reference . . . . .	981
8.1008.1Detailed Description . . . . .	981
8.1008.2Field Documentation . . . . .	981
8.1008.2.1switchOption . . . . .	981
8.1009VoiceAnswerCall Struct Reference . . . . .	981
8.1009.1Detailed Description . . . . .	981
8.1009.2Field Documentation . . . . .	982
8.1009.2.1pCallId . . . . .	982
8.1010VoiceBindSubscriptionInfo Struct Reference . . . . .	982
8.1010.1Detailed Description . . . . .	982

8.1010.2Field Documentation . . . . .	982
8.1010.2.1subsType . . . . .	982
8.1011VoiceBurstDTMFInfo Struct Reference . . . . .	982
8.1011.1Detailed Description . . . . .	982
8.1011.2Field Documentation . . . . .	982
8.1011.2.1BurstDTMFInfo . . . . .	982
8.1011.2.2BurstDTMFLengths . . . . .	983
8.1012VoiceCallInfoReq Struct Reference . . . . .	983
8.1012.1Detailed Description . . . . .	983
8.1012.2Field Documentation . . . . .	983
8.1012.2.1callID . . . . .	983
8.1013VoiceCallInfoResp Struct Reference . . . . .	983
8.1013.1Detailed Description . . . . .	983
8.1013.2Field Documentation . . . . .	985
8.1013.2.1pAlertingPattern . . . . .	985
8.1013.2.2pAlertType . . . . .	985
8.1013.2.3pAlphaIDInfo . . . . .	985
8.1013.2.4pCallInfo . . . . .	985
8.1013.2.5pConnectNumInfo . . . . .	985
8.1013.2.6pDiagInfo . . . . .	985
8.1013.2.7pOTASPStatus . . . . .	985
8.1013.2.8pRemotePartyName . . . . .	985
8.1013.2.9pRemotePartyNum . . . . .	985
8.1013.2.10pSrvOpt . . . . .	985
8.1013.2.11pUUSInfo . . . . .	986
8.1013.2.12pVoicePrivacy . . . . .	986
8.1014VoiceCallRequestParams Struct Reference . . . . .	986
8.1014.1Detailed Description . . . . .	986
8.1014.2Field Documentation . . . . .	987
8.1014.2.1callNumber . . . . .	987
8.1014.2.2pCallPartySubAdd . . . . .	987
8.1014.2.3pCallType . . . . .	987
8.1014.2.4pCLIRType . . . . .	987
8.1014.2.5pCUGInfo . . . . .	987
8.1014.2.6pEmergencyCategory . . . . .	987
8.1014.2.7pSvcType . . . . .	987
8.1014.2.8pUUSInfo . . . . .	987
8.1015VoiceCallResponseParams Struct Reference . . . . .	987
8.1015.1Detailed Description . . . . .	988
8.1015.2Field Documentation . . . . .	988

8.1015.2.1pAlphaIDInfo . . . . .	988
8.1015.2.2pCallID . . . . .	988
8.1015.2.3pCCResultType . . . . .	988
8.1015.2.4pCCSUPSType . . . . .	988
8.1016VoiceContDTMFinfo Struct Reference . . . . .	988
8.1016.1Detailed Description . . . . .	988
8.1016.2Field Documentation . . . . .	989
8.1016.2.1DTMFdigit . . . . .	989
8.1016.2.2pCallID . . . . .	989
8.1017VoiceDTMFEventInfo Struct Reference . . . . .	989
8.1017.1Detailed Description . . . . .	989
8.1017.2Field Documentation . . . . .	990
8.1017.2.1DTMFInformation . . . . .	990
8.1017.2.2pOffLength . . . . .	990
8.1017.2.3pOnLength . . . . .	990
8.1018VoiceFlashInfo Struct Reference . . . . .	990
8.1018.1Detailed Description . . . . .	990
8.1018.2Field Documentation . . . . .	990
8.1018.2.1pCallID . . . . .	990
8.1018.2.2pFlashPayLd . . . . .	990
8.1018.2.3pFlashType . . . . .	990
8.1019VoiceGetAllCallInfo Struct Reference . . . . .	990
8.1019.1Detailed Description . . . . .	991
8.1019.2Field Documentation . . . . .	993
8.1019.2.1pArrAlertingPattern . . . . .	993
8.1019.2.2pArrAlertingType . . . . .	993
8.1019.2.3pArrAlphaID . . . . .	993
8.1019.2.4pArrCalledPartyNum . . . . .	993
8.1019.2.5pArrCallEndReason . . . . .	993
8.1019.2.6pArrCallInfo . . . . .	993
8.1019.2.7pArrConnectPartyNum . . . . .	993
8.1019.2.8pArrDiagInfo . . . . .	993
8.1019.2.9pArrRedirPartyNum . . . . .	993
8.1019.2.10pArrRemotePartyName . . . . .	993
8.1019.2.11pArrRemotePartyNum . . . . .	993
8.1019.2.12pArrSvcOption . . . . .	993
8.1019.2.13pArrUUSInfo . . . . .	993
8.1019.2.14pOTASPSStatus . . . . .	993
8.1019.2.15pVoicePrivacy . . . . .	993
8.1020VoiceGetCallBarringReq Struct Reference . . . . .	993

8.1020.1 Detailed Description . . . . .	993
8.1020.2 Field Documentation . . . . .	994
8.1020.2.1 pSvcClass . . . . .	994
8.1020.2.2 reason . . . . .	994
8.1021 VoiceGetCallBarringResp Struct Reference . . . . .	994
8.1021.1 Detailed Description . . . . .	994
8.1021.2 Field Documentation . . . . .	995
8.1021.2.1 pAlphaIDInfo . . . . .	995
8.1021.2.2 pCallID . . . . .	995
8.1021.2.3 pCCResType . . . . .	995
8.1021.2.4 pCCSUPSType . . . . .	995
8.1021.2.5 pFailCause . . . . .	995
8.1021.2.6 pSvcClass . . . . .	995
8.1022 VoiceGetCallFWReq Struct Reference . . . . .	995
8.1022.1 Detailed Description . . . . .	996
8.1022.2 Field Documentation . . . . .	996
8.1022.2.1 pSvcClass . . . . .	996
8.1022.2.2 Reason . . . . .	996
8.1023 VoiceGetCallFWResp Struct Reference . . . . .	996
8.1023.1 Detailed Description . . . . .	996
8.1023.2 Field Documentation . . . . .	997
8.1023.2.1 pAlphaIDInfo . . . . .	997
8.1023.2.2 pCallID . . . . .	997
8.1023.2.3 pCCResType . . . . .	997
8.1023.2.4 pCCSUPSType . . . . .	997
8.1023.2.5 pFailCause . . . . .	997
8.1023.2.6 pGetCallFWExtInfo . . . . .	997
8.1023.2.7 pGetCallFWInfo . . . . .	997
8.1024 VoiceGetCallWaitInfo Struct Reference . . . . .	997
8.1024.1 Detailed Description . . . . .	998
8.1024.2 Field Documentation . . . . .	998
8.1024.2.1 pAlphaIDInfo . . . . .	999
8.1024.2.2 pCallID . . . . .	999
8.1024.2.3 pCCResType . . . . .	999
8.1024.2.4 pCCSUPSType . . . . .	999
8.1024.2.5 pFailCause . . . . .	999
8.1024.2.6 pSvcClass . . . . .	999
8.1025 VoiceGetCLIPResp Struct Reference . . . . .	999
8.1025.1 Detailed Description . . . . .	999
8.1025.2 Field Documentation . . . . .	1000

8.1025.2.1pAlphaIDInfo . . . . .	1000
8.1025.2.2pCallID . . . . .	1000
8.1025.2.3pCCResType . . . . .	1000
8.1025.2.4pCCSUPSType . . . . .	1000
8.1025.2.5pCLIPResp . . . . .	1000
8.1025.2.6pFailCause . . . . .	1000
8.1026VoiceGetCLIRResp Struct Reference . . . . .	1000
8.1026.1Detailed Description . . . . .	1000
8.1026.2Field Documentation . . . . .	1001
8.1026.2.1pAlphaIDInfo . . . . .	1001
8.1026.2.2pCallID . . . . .	1001
8.1026.2.3pCCResType . . . . .	1001
8.1026.2.4pCCSUPSType . . . . .	1001
8.1026.2.5pCLIRResp . . . . .	1001
8.1026.2.6pFailCause . . . . .	1001
8.1027VoiceGetCNAPResp Struct Reference . . . . .	1001
8.1027.1Detailed Description . . . . .	1001
8.1027.2Field Documentation . . . . .	1002
8.1027.2.1pAlphaIDInfo . . . . .	1002
8.1027.2.2pCallID . . . . .	1002
8.1027.2.3pCCResType . . . . .	1002
8.1027.2.4pCCSUPSType . . . . .	1002
8.1027.2.5pCNAPResp . . . . .	1002
8.1027.2.6pFailCause . . . . .	1002
8.1028VoiceGetCOLPResp Struct Reference . . . . .	1002
8.1028.1Detailed Description . . . . .	1003
8.1028.2Field Documentation . . . . .	1003
8.1028.2.1pAlphaIDInfo . . . . .	1003
8.1028.2.2pCallID . . . . .	1004
8.1028.2.3pCCResType . . . . .	1004
8.1028.2.4pCCSUPSType . . . . .	1004
8.1028.2.5pCOLPResp . . . . .	1004
8.1028.2.6pFailCause . . . . .	1004
8.1029VoiceGetCOLRResp Struct Reference . . . . .	1004
8.1029.1Detailed Description . . . . .	1004
8.1029.2Field Documentation . . . . .	1005
8.1029.2.1pAlphaIDInfo . . . . .	1005
8.1029.2.2pCallID . . . . .	1005
8.1029.2.3pCCResType . . . . .	1005
8.1029.2.4pCCSUPSType . . . . .	1005

8.1029.2.5pCOLRResp . . . . .	1005
8.1029.2.6pFailCause . . . . .	1005
8.1030VoiceGetConfigReq Struct Reference . . . . .	1005
8.1030.1Detailed Description . . . . .	1005
8.1030.2Field Documentation . . . . .	1006
8.1030.2.1pAirTimer . . . . .	1006
8.1030.2.2pAMRStatus . . . . .	1006
8.1030.2.3pAutoAnswer . . . . .	1006
8.1030.2.4pNamID . . . . .	1006
8.1030.2.5pPrefVoicePrivacy . . . . .	1006
8.1030.2.6pPrefVoiceSO . . . . .	1006
8.1030.2.7pRoamTimer . . . . .	1007
8.1030.2.8pTTYMode . . . . .	1007
8.1030.2.9pVoiceDomainPref . . . . .	1007
8.1031VoiceGetConfigResp Struct Reference . . . . .	1007
8.1031.1Detailed Description . . . . .	1007
8.1031.2Field Documentation . . . . .	1008
8.1031.2.1pAirTimerCnt . . . . .	1008
8.1031.2.2pAutoAnswerStat . . . . .	1008
8.1031.2.3pCurAMRConfig . . . . .	1008
8.1031.2.4pCurPrefVoiceSO . . . . .	1008
8.1031.2.5pCurrTTYMode . . . . .	1008
8.1031.2.6pCurVoiceDomainPref . . . . .	1008
8.1031.2.7pCurVoicePrivacyPref . . . . .	1008
8.1031.2.8pRoamTimerCnt . . . . .	1008
8.1032VoiceIndicationRegisterInfo Struct Reference . . . . .	1008
8.1032.1Detailed Description . . . . .	1009
8.1032.2Field Documentation . . . . .	1009
8.1032.2.1pRegDTMFEvents . . . . .	1009
8.1032.2.2pRegVoicePrivacyEvents . . . . .	1009
8.1032.2.3pSuppsNotifEvents . . . . .	1009
8.1033VoiceInfoRec Struct Reference . . . . .	1009
8.1033.1Detailed Description . . . . .	1010
8.1033.2Field Documentation . . . . .	1011
8.1033.2.1callID . . . . .	1011
8.1033.2.2pCalledPartyInfo . . . . .	1011
8.1033.2.3pCallerIDInfo . . . . .	1011
8.1033.2.4pCallerNameInfo . . . . .	1011
8.1033.2.5pCallingPartyInfo . . . . .	1011
8.1033.2.6pCallWaitInd . . . . .	1011

8.1033.2.7pCLIRCause . . . . .	1011
8.1033.2.8pConnectNumInfo . . . . .	1011
8.1033.2.9pDispInfo . . . . .	1011
8.1033.2.10pExtDispInfo . . . . .	1011
8.1033.2.11pExtDispRecInfo . . . . .	1011
8.1033.2.12pLineCtrlInfo . . . . .	1011
8.1033.2.13pNSSAudioCtrl . . . . .	1011
8.1033.2.14pNSSRelease . . . . .	1011
8.1033.2.15pRedirNumInfo . . . . .	1012
8.1033.2.16pSignalInfo . . . . .	1012
8.1034VoiceManageCallsReq Struct Reference . . . . .	1012
8.1034.1Detailed Description . . . . .	1012
8.1034.2Field Documentation . . . . .	1012
8.1034.2.1pCallID . . . . .	1012
8.1034.2.2pSUPSType . . . . .	1012
8.1035VoiceManageCallsResp Struct Reference . . . . .	1012
8.1035.1Detailed Description . . . . .	1013
8.1035.2Field Documentation . . . . .	1013
8.1035.2.1pFailCause . . . . .	1013
8.1036VoiceOrigUSSDNoWaitInfo Struct Reference . . . . .	1013
8.1036.1Detailed Description . . . . .	1013
8.1036.2Field Documentation . . . . .	1013
8.1036.2.1USSInformation . . . . .	1013
8.1037VoiceOTASPStatusInfo Struct Reference . . . . .	1013
8.1037.1Detailed Description . . . . .	1013
8.1037.2Field Documentation . . . . .	1014
8.1037.2.1callID . . . . .	1014
8.1037.2.2OTASPStatus . . . . .	1014
8.1038VoicePrivacyInfo Struct Reference . . . . .	1014
8.1038.1Detailed Description . . . . .	1014
8.1038.2Field Documentation . . . . .	1015
8.1038.2.1callID . . . . .	1015
8.1038.2.2voicePrivacy . . . . .	1015
8.1039VoiceSetAllCallStatusCbkJInfo Struct Reference . . . . .	1015
8.1039.1Detailed Description . . . . .	1015
8.1039.2Field Documentation . . . . .	1017
8.1039.2.1arrCallInformation . . . . .	1017
8.1039.2.2pArrAlertingPattern . . . . .	1017
8.1039.2.3pArrAlertingType . . . . .	1017
8.1039.2.4pArrAlphaID . . . . .	1017

8.1039.2.5pArrCalledPartyNum . . . . .	1017
8.1039.2.6pArrCallEndReason . . . . .	1017
8.1039.2.7pArrConnectPartyNum . . . . .	1017
8.1039.2.8pArrDiagInfo . . . . .	1017
8.1039.2.9pArrRedirPartyNum . . . . .	1017
8.1039.2.10pArrRemotePartyName . . . . .	1017
8.1039.2.11pArrRemotePartyNum . . . . .	1017
8.1039.2.12pArrSvcOption . . . . .	1017
8.1040VoiceSetCallBarringPwdInfo Struct Reference . . . . .	1017
8.1040.1Detailed Description . . . . .	1017
8.1040.2Field Documentation . . . . .	1018
8.1040.2.1newPasswd . . . . .	1018
8.1040.2.2newPasswdAgain . . . . .	1018
8.1040.2.3oldPasswd . . . . .	1018
8.1040.2.4Reason . . . . .	1018
8.1041VoiceSetCallBarringPwdResp Struct Reference . . . . .	1018
8.1041.1Detailed Description . . . . .	1019
8.1041.2Field Documentation . . . . .	1019
8.1041.2.1pAlphaIDInfo . . . . .	1019
8.1041.2.2pCallID . . . . .	1019
8.1041.2.3pCCResType . . . . .	1019
8.1041.2.4pCCSUPSType . . . . .	1019
8.1041.2.5pFailCause . . . . .	1019
8.1042VoiceSetConfigReq Struct Reference . . . . .	1019
8.1042.1Detailed Description . . . . .	1020
8.1042.2Field Documentation . . . . .	1020
8.1042.2.1pAirTimerConfig . . . . .	1021
8.1042.2.2pAutoAnswer . . . . .	1021
8.1042.2.3pPrefVoiceDomain . . . . .	1021
8.1042.2.4pPrefVoiceSO . . . . .	1021
8.1042.2.5pRoamTimerConfig . . . . .	1021
8.1042.2.6pTTYMode . . . . .	1021
8.1043VoiceSetConfigResp Struct Reference . . . . .	1021
8.1043.1Detailed Description . . . . .	1021
8.1043.2Field Documentation . . . . .	1022
8.1043.2.1pAirTimerStatus . . . . .	1022
8.1043.2.2pAutoAnsStatus . . . . .	1022
8.1043.2.3pPrefVoiceSOStatus . . . . .	1022
8.1043.2.4pRoamTimerStatus . . . . .	1022
8.1043.2.5pTTYConfigStatus . . . . .	1022

8.1043.2.6pVoiceDomainPrefStatus . . . . .	1022
8.1044VoiceSetPrefPrivacy Struct Reference . . . . .	1022
8.1044.1Detailed Description . . . . .	1022
8.1044.2Field Documentation . . . . .	1023
8.1044.2.1privacyPref . . . . .	1023
8.1045VoiceSetSUPSServiceReq Struct Reference . . . . .	1023
8.1045.1Detailed Description . . . . .	1023
8.1045.2Field Documentation . . . . .	1025
8.1045.2.1pCallBarringPasswd . . . . .	1025
8.1045.2.2pCallForwardingNumber . . . . .	1025
8.1045.2.3pCallFwdTypeAndPlan . . . . .	1025
8.1045.2.4pServiceClass . . . . .	1025
8.1045.2.5pTimerVal . . . . .	1025
8.1045.2.6reason . . . . .	1025
8.1045.2.7voiceSvc . . . . .	1025
8.1046VoiceSetSUPSServiceResp Struct Reference . . . . .	1025
8.1046.1Detailed Description . . . . .	1025
8.1046.2Field Documentation . . . . .	1026
8.1046.2.1pAlphaIDInfo . . . . .	1026
8.1046.2.2pCallID . . . . .	1026
8.1046.2.3pCCResultType . . . . .	1026
8.1046.2.4pCCSUPSType . . . . .	1026
8.1046.2.5pFailCause . . . . .	1026
8.1047VoiceStopContDTMFInfo Struct Reference . . . . .	1026
8.1047.1Detailed Description . . . . .	1026
8.1047.2Field Documentation . . . . .	1027
8.1047.2.1callID . . . . .	1027
8.1048VoiceSUPSInfo Struct Reference . . . . .	1027
8.1048.1Detailed Description . . . . .	1027
8.1048.2Field Documentation . . . . .	1029
8.1048.2.1pAlphaIDInfo . . . . .	1029
8.1048.2.2pCallBarPasswd . . . . .	1029
8.1048.2.3pCallFwdInfo . . . . .	1029
8.1048.2.4pCallFWNum . . . . .	1029
8.1048.2.5pCallFWTimerVal . . . . .	1029
8.1048.2.6pCallID . . . . .	1029
8.1048.2.7pCLIPstatus . . . . .	1029
8.1048.2.8pCLIRstatus . . . . .	1029
8.1048.2.9pCNAPstatus . . . . .	1029
8.1048.2.10pCOLPstatus . . . . .	1029

8.1048.2.1	COLRstatus	1029
8.1048.2.2	DataSrc	1029
8.1048.2.3	FailCause	1029
8.1048.2.4	NewPwdData	1029
8.1048.2.5	Reason	1029
8.1048.2.6	SvcClass	1029
8.1048.2.7	USSInfo	1029
8.1048.2.8	SUPSInformation	1029
8.1049	oiceSUPSNotification Struct Reference	1029
8.1049.1	Detailed Description	1030
8.1049.2	Field Documentation	1031
8.1049.2.1	callID	1031
8.1049.2.2	notifType	1031
8.1049.2.3	CUGIndex	1031
8.1049.2.4	ECTNum	1031
8.1050	wcdmaCellInfo Struct Reference	1031
8.1050.1	Detailed Description	1031
8.1050.2	Field Documentation	1031
8.1050.2.1	cpich_ecno	1031
8.1050.2.2	cpich_rscp	1031
8.1050.2.3	psc	1032
8.1050.2.4	srlev	1032
8.1051	WCDMAECIOTresh Struct Reference	1032
8.1051.1	Detailed Description	1032
8.1051.2	Field Documentation	1032
8.1051.2.1	pWCDMAECIOTreshList	1032
8.1051.2.2	WCDMAECIOTreshListLen	1032
8.1052	WCDMAInfoLTENNeighborCell Struct Reference	1032
8.1052.1	Detailed Description	1032
8.1052.2	Field Documentation	1033
8.1052.2.1	UMTSLTENbrCell	1033
8.1052.2.2	umtsLTENbrCellLen	1033
8.1052.2.3	wcdmaRRCState	1033
8.1053	wcdmaLongMsgDecodingParams Struct Reference	1033
8.1053.1	Detailed Description	1033
8.1053.2	Field Documentation	1034
8.1053.2.1	Date	1034
8.1053.2.2	IsUDHPresent	1034
8.1053.2.3	Message	1035
8.1053.2.4	PartNum	1035

8.1053.2.5pReferenceNum . . . . .	1035
8.1053.2.6pScAddr . . . . .	1035
8.1053.2.7pScAddrLength . . . . .	1035
8.1053.2.8pSenderAddr . . . . .	1035
8.1053.2.9pSenderAddrLength . . . . .	1035
8.1053.2.10pTextMsg . . . . .	1035
8.1053.2.11pTextMsgLength . . . . .	1035
8.1053.2.12pTotalNum . . . . .	1035
8.1053.2.13pTime . . . . .	1035
8.1054wcdmaMsgDecodingParams Struct Reference . . . . .	1035
8.1054.1Detailed Description . . . . .	1035
8.1054.2Field Documentation . . . . .	1036
8.1054.2.1Date . . . . .	1036
8.1054.2.2pMessage . . . . .	1036
8.1054.2.3pScAddr . . . . .	1036
8.1054.2.4pScAddrLength . . . . .	1036
8.1054.2.5pSenderAddr . . . . .	1036
8.1054.2.6pSenderAddrLength . . . . .	1036
8.1054.2.7pTextMsg . . . . .	1036
8.1054.2.8pTextMsgLength . . . . .	1036
8.1054.2.9pTime . . . . .	1036
8.1055wcdmaMsgEncodingParams Struct Reference . . . . .	1036
8.1055.1Detailed Description . . . . .	1037
8.1055.2Field Documentation . . . . .	1037
8.1055.2.1alphabet . . . . .	1037
8.1055.2.2messageSize . . . . .	1037
8.1055.2.3pDestAddr . . . . .	1037
8.1055.2.4pPDUMessage . . . . .	1037
8.1055.2.5pTextMsg . . . . .	1037
8.1056wCDMARSSIThresh Struct Reference . . . . .	1037
8.1056.1Detailed Description . . . . .	1037
8.1056.2Field Documentation . . . . .	1038
8.1056.2.1pWCDMARSSIThreshList . . . . .	1038
8.1056.2.2wCDMARSSIThreshListLen . . . . .	1038
8.1057wCDMASysInfo Struct Reference . . . . .	1038
8.1057.1Detailed Description . . . . .	1038
8.1057.2Field Documentation . . . . .	1040
8.1057.2.1cellId . . . . .	1041
8.1057.2.2cellIdValid . . . . .	1041
8.1057.2.3hscCallStatus . . . . .	1041

8.1057.2.4hsCallStatusValid . . . . .	1041
8.1057.2.5hsInd . . . . .	1041
8.1057.2.6hsIndValid . . . . .	1041
8.1057.2.7lac . . . . .	1041
8.1057.2.8acValid . . . . .	1041
8.1057.2.9MCC . . . . .	1041
8.1057.2.10INC . . . . .	1041
8.1057.2.11networkIdValid . . . . .	1041
8.1057.2.12sc . . . . .	1041
8.1057.2.13scValid . . . . .	1041
8.1057.2.14egRejectInfoValid . . . . .	1041
8.1057.2.15jCause . . . . .	1041
8.1057.2.16jectSrvDomain . . . . .	1041
8.1057.2.17sysInfoWCDMA . . . . .	1041
8.1058wcdmaUARFCN Struct Reference . . . . .	1041
8.1058.1Detailed Description . . . . .	1041
8.1058.2Field Documentation . . . . .	1042
8.1058.2.1status . . . . .	1042
8.1058.2.2uarfcn . . . . .	1042
8.1059nds_currNetworkInfo Struct Reference . . . . .	1042
8.1059.1Detailed Description . . . . .	1042
8.1059.2Field Documentation . . . . .	1043
8.1059.2.1NetworkType . . . . .	1043
8.1059.2.2RATMask . . . . .	1043
8.1059.2.3SOMask . . . . .	1043
8.1060nds_Domain Struct Reference . . . . .	1043
8.1060.1Detailed Description . . . . .	1043
8.1060.2Field Documentation . . . . .	1044
8.1060.2.1domainLen . . . . .	1044
8.1060.2.2domainName . . . . .	1044
8.1061nds_DomainNameList Struct Reference . . . . .	1044
8.1061.1Detailed Description . . . . .	1044
8.1061.2Field Documentation . . . . .	1044
8.1061.2.1domain . . . . .	1044
8.1061.2.2numInstances . . . . .	1044
8.1062nds_GPRSQoS Struct Reference . . . . .	1044
8.1062.1Detailed Description . . . . .	1044
8.1062.2Field Documentation . . . . .	1045
8.1062.2.1delayClass . . . . .	1045
8.1062.2.2meanThroughputClass . . . . .	1045

8.1062.2.3	peakThroughputClass	1045
8.1062.2.4	precedenceClass	1045
8.1062.2.5	reliabilityClass	1045
8.1063	nds_IPV6AddressInfo Struct Reference	1045
8.1063.1	Detailed Description	1045
8.1063.2	Field Documentation	1045
8.1063.2.1	IPAddressV6	1045
8.1063.2.2	IPV6PrefixLen	1045
8.1064	nds_IPV6GWAddressInfo Struct Reference	1045
8.1064.1	Detailed Description	1046
8.1064.2	Field Documentation	1046
8.1064.2.1	gwAddressV6	1046
8.1064.2.2	gwV6PrefixLen	1046
8.1065	nds_PCSCFFQDNAddress Struct Reference	1046
8.1065.1	Detailed Description	1046
8.1065.2	Field Documentation	1046
8.1065.2.1	fqdnAddr	1046
8.1065.2.2	fqdnLen	1046
8.1066	nds_PCSCFFQDNAddressList Struct Reference	1046
8.1066.1	Detailed Description	1047
8.1066.2	Field Documentation	1047
8.1066.2.1	numInstances	1047
8.1066.2.2	pcsfQDNAddress	1047
8.1067	nds_PCSCFIPv4ServerAddressList Struct Reference	1047
8.1067.1	Detailed Description	1047
8.1067.2	Field Documentation	1047
8.1067.2.1	numInstances	1047
8.1067.2.2	pcsfIPv4Addr	1047
8.1068	nds_ProfileIdentifier Struct Reference	1047
8.1068.1	Detailed Description	1048
8.1068.2	Field Documentation	1048
8.1068.2.1	profileIndex	1048
8.1068.2.2	profileType	1048
8.1069	nds_profileInfo Union Reference	1048
8.1069.1	Detailed Description	1048
8.1069.2	Field Documentation	1048
8.1069.2.1	SIqsProfile3GPP	1048
8.1069.2.2	SIqsProfile3GPP2	1048
8.1070	nds_UMTSMinQoS Struct Reference	1048
8.1070.1	Detailed Description	1049

8.1070.2Field Documentation . . . . .	1050
8.1070.2.1deliveryErrSDU . . . . .	1050
8.1070.2.2grntDownlinkBitrate . . . . .	1050
8.1070.2.3grntUplinkBitrate . . . . .	1050
8.1070.2.4maxDownlinkBitrate . . . . .	1050
8.1070.2.5maxSDUSize . . . . .	1050
8.1070.2.6maxUplinkBitrate . . . . .	1050
8.1070.2.7qosDeliveryOrder . . . . .	1050
8.1070.2.8resBerRatio . . . . .	1051
8.1070.2.9sduErrorRatio . . . . .	1051
8.1070.2.10trafficClass . . . . .	1051
8.1070.2.11trafficPriority . . . . .	1051
8.1070.2.12transferDelay . . . . .	1051
8.1071WdsByteTotals Struct Reference . . . . .	1051
8.1071.1Detailed Description . . . . .	1051
8.1071.2Field Documentation . . . . .	1051
8.1071.2.1ByteTotalsElmntsV4 . . . . .	1051
8.1071.2.2ByteTotalsElmntsV6 . . . . .	1051
8.1071.2.3pV4sessionId . . . . .	1051
8.1071.2.4pV6sessionId . . . . .	1051
8.1072WdsByteTotalsElmnts Struct Reference . . . . .	1052
8.1072.1Detailed Description . . . . .	1052
8.1072.2Field Documentation . . . . .	1052
8.1072.2.1pRXTotalBytes . . . . .	1052
8.1072.2.2pTXTotalBytes . . . . .	1052
8.1073WdsClientLeaseChange Struct Reference . . . . .	1052
8.1073.1Detailed Description . . . . .	1052
8.1073.2Field Documentation . . . . .	1052
8.1073.2.1pEnableNotification . . . . .	1052
8.1074WdsConnectionRate Struct Reference . . . . .	1052
8.1074.1Detailed Description . . . . .	1053
8.1074.2Field Documentation . . . . .	1053
8.1074.2.1ConnRateElmntsV4 . . . . .	1053
8.1074.2.2ConnRateElmntsV6 . . . . .	1053
8.1074.2.3pV4sessionId . . . . .	1053
8.1074.2.4pV6sessionId . . . . .	1053
8.1075WdsConnectionRateElmnts Struct Reference . . . . .	1053
8.1075.1Detailed Description . . . . .	1053
8.1075.2Field Documentation . . . . .	1054
8.1075.2.1pCurrentChannelRXRate . . . . .	1054

8.1075.2.2pCurrentChannelTXRate . . . . .	1054
8.1075.2.3pMaxChannelRXRate . . . . .	1054
8.1075.2.4pMaxChannelTXRate . . . . .	1054
8.1076WdsDHCPv4ClientLeaseInd Struct Reference . . . . .	1054
8.1076.1Detailed Description . . . . .	1054
8.1076.2Field Documentation . . . . .	1055
8.1076.2.1pIPv4Addr . . . . .	1055
8.1076.2.2pLeaseState . . . . .	1055
8.1076.2.3pOptList . . . . .	1055
8.1076.2.4pProfileId . . . . .	1055
8.1077WdsDHCPv4Config Struct Reference . . . . .	1055
8.1077.1Detailed Description . . . . .	1055
8.1077.2Field Documentation . . . . .	1056
8.1077.2.1pHwConfig . . . . .	1056
8.1077.2.2pProfileId . . . . .	1056
8.1077.2.3pRequestOptionList . . . . .	1056
8.1078WdsDhcpv4HwConfig Struct Reference . . . . .	1056
8.1078.1Detailed Description . . . . .	1056
8.1078.2Field Documentation . . . . .	1056
8.1078.2.1chaddr . . . . .	1056
8.1078.2.2chaddrLen . . . . .	1056
8.1078.2.3hwType . . . . .	1056
8.1079WdsDHCPv4HWConfig Struct Reference . . . . .	1056
8.1079.1Detailed Description . . . . .	1057
8.1079.2Field Documentation . . . . .	1057
8.1079.2.1chaddr . . . . .	1057
8.1079.2.2chaddrLen . . . . .	1057
8.1079.2.3hwType . . . . .	1057
8.1080WdsDHCPv4Option Struct Reference . . . . .	1057
8.1080.1Detailed Description . . . . .	1057
8.1080.2Field Documentation . . . . .	1057
8.1080.2.1optCode . . . . .	1057
8.1080.2.2optVal . . . . .	1058
8.1080.2.3optValLen . . . . .	1058
8.1081WdsDhcpv4Option Struct Reference . . . . .	1058
8.1081.1Detailed Description . . . . .	1058
8.1081.2Field Documentation . . . . .	1058
8.1081.2.1optCode . . . . .	1058
8.1081.2.2optVal . . . . .	1058
8.1081.2.3optValLen . . . . .	1058

8.1082	<a href="#">WdsDhcpv4OptionList Struct Reference</a>	1058
8.1082.1	<a href="#">Detailed Description</a>	1058
8.1082.2	<a href="#">Field Documentation</a>	1058
8.1082.2.1	<a href="#">numOpt</a>	1058
8.1082.2.2	<a href="#">pOptList</a>	1059
8.1083	<a href="#">WdsDHCPv4OptionList Struct Reference</a>	1059
8.1083.1	<a href="#">Detailed Description</a>	1059
8.1083.2	<a href="#">Field Documentation</a>	1059
8.1083.2.1	<a href="#">numOpt</a>	1059
8.1083.2.2	<a href="#">pOptList</a>	1059
8.1084	<a href="#">WdsDHCPv4ProfileId Struct Reference</a>	1059
8.1084.1	<a href="#">Detailed Description</a>	1059
8.1084.2	<a href="#">Field Documentation</a>	1059
8.1084.2.1	<a href="#">profileId</a>	1060
8.1084.2.2	<a href="#">profileType</a>	1060
8.1085	<a href="#">WdsDhcpv4ProfileId Struct Reference</a>	1060
8.1085.1	<a href="#">Detailed Description</a>	1060
8.1085.2	<a href="#">Field Documentation</a>	1060
8.1085.2.1	<a href="#">profileId</a>	1060
8.1085.2.2	<a href="#">profileType</a>	1060
8.1086	<a href="#">WDSGetLoopbackData Struct Reference</a>	1060
8.1086.1	<a href="#">Detailed Description</a>	1060
8.1086.2	<a href="#">Field Documentation</a>	1061
8.1086.2.1	<a href="#">ByteLoopbackMode</a>	1061
8.1086.2.2	<a href="#">ByteLoopbackMultiplier</a>	1061
8.1087	<a href="#">WdsIpAddressInfoReq Struct Reference</a>	1061
8.1087.1	<a href="#">Field Documentation</a>	1061
8.1087.1.1	<a href="#">ip</a>	1061
8.1087.1.2	<a href="#">pv4sessionId</a>	1061
8.1087.1.3	<a href="#">pv6sessionId</a>	1061
8.1088	<a href="#">WdsPktStatisticsElmnts Struct Reference</a>	1061
8.1088.1	<a href="#">Detailed Description</a>	1062
8.1088.2	<a href="#">Field Documentation</a>	1062
8.1088.2.1	<a href="#">pRXDroppedCount</a>	1062
8.1088.2.2	<a href="#">pRXOkBytesCount</a>	1062
8.1088.2.3	<a href="#">pRXOKBytesLastCall</a>	1062
8.1088.2.4	<a href="#">pRXPacketErrors</a>	1063
8.1088.2.5	<a href="#">pRXPacketOverflows</a>	1063
8.1088.2.6	<a href="#">pRXPacketSuccesses</a>	1063
8.1088.2.7	<a href="#">pTXDroppedCount</a>	1063

8.1088.2.8pTXOkBytesCount . . . . .	1063
8.1088.2.9pTXOKBytesLastCall . . . . .	1063
8.1088.2.10pTXPacketErrors . . . . .	1063
8.1088.2.11pTXPacketOverflows . . . . .	1063
8.1088.2.12pTXPacketSuccesses . . . . .	1063
8.1089WdsPktStatisticsReq Struct Reference . . . . .	1063
8.1089.1Detailed Description . . . . .	1063
8.1089.2Field Documentation . . . . .	1063
8.1089.2.1pStatMask . . . . .	1063
8.1090WdsPktStatisticsResp Struct Reference . . . . .	1063
8.1090.1Detailed Description . . . . .	1063
8.1090.2Field Documentation . . . . .	1064
8.1090.2.1PktStatElmntsV4 . . . . .	1064
8.1090.2.2PktStatElmntsV6 . . . . .	1064
8.1090.2.3pV4sessionId . . . . .	1064
8.1090.2.4pV6sessionId . . . . .	1064
8.1091WdsProfileParam Union Reference . . . . .	1064
8.1091.1Detailed Description . . . . .	1064
8.1091.2Field Documentation . . . . .	1064
8.1091.2.1SlqsProfile3GPP . . . . .	1064
8.1091.2.2SlqsProfile3GPP2 . . . . .	1064
8.1092WdsRunTimeSettings Struct Reference . . . . .	1064
8.1092.1Detailed Description . . . . .	1065
8.1092.2Field Documentation . . . . .	1065
8.1092.2.1rts . . . . .	1065
8.1092.2.2v4sessionId . . . . .	1065
8.1092.2.3v6sessionId . . . . .	1065
8.1093WdsSetEventReportReq Struct Reference . . . . .	1065
8.1093.1Detailed Description . . . . .	1065
8.1093.2Field Documentation . . . . .	1067
8.1093.2.1pCurrChannelRateInd . . . . .	1067
8.1093.2.2pCurrDataBearerTechInd . . . . .	1067
8.1093.2.3pCurrPrefDataSysInd . . . . .	1067
8.1093.2.4pDataBearerTechInd . . . . .	1067
8.1093.2.5pDataCallStatusChangeInd . . . . .	1067
8.1093.2.6pDataSystemStatusChangeInd . . . . .	1067
8.1093.2.7pDormancyStatusInd . . . . .	1067
8.1093.2.8pEVDOPageMonPerChangeInd . . . . .	1067
8.1093.2.9pMIPStatusInd . . . . .	1067
8.1093.2.10pTransferStatInd . . . . .	1067

8.1094	WDSSetLoopbackData Struct Reference	1067
8.1094.1	Detailed Description	1067
8.1094.2	Field Documentation	1067
8.1094.2.1	LoopbackMode	1067
8.1094.2.2	LoopbackMultiplier	1067
8.1095	WDSSWICurrentChannelRates Struct Reference	1067
8.1095.1	Detailed Description	1068
8.1095.2	Field Documentation	1068
8.1095.2.1	current_channel_rx_rate	1068
8.1095.2.2	current_channel_tx_rate	1068
8.1095.2.3	max_channel_rx_rate	1068
8.1095.2.4	max_channel_tx_rate	1068
<b>9</b>	<b>File Documentation</b>	<b>1069</b>
9.1	apdoxypages.c File Reference	1069
9.1.1	Detailed Description	1069
9.2	common.h File Reference	1069
9.2.1	Macro Definition Documentation	1071
9.2.1.1	DEFAULT_LOC_TIMEOUT_IN_SEC	1071
9.2.1.2	MINREQBKLEN	1071
9.2.1.3	MSGID_AND_LEN	1071
9.2.1.4	MSGID_DONT_CARE	1071
9.2.1.5	SDK_VALIDATE_INPUT_PACK_PARAM	1071
9.2.1.6	SDU_HDR_LEN	1071
9.2.1.7	UNUSEDPARAM	1071
9.2.2	Typedef Documentation	1071
9.2.2.1	logger	1071
9.2.3	Enumeration Type Documentation	1071
9.2.3.1	eLOG_LEVEL	1071
9.2.3.2	eQMI_SVC	1071
9.2.3.3	eTimeout	1072
9.2.3.4	msgtype	1072
9.2.4	Function Documentation	1072
9.2.4.1	fill_pack_ctx	1072
9.2.4.2	fill_sdu_hdr	1072
9.2.4.3	get_version	1072
9.2.4.4	helper_get_resp_ctx	1072
9.2.4.5	helper_get_xid	1073
9.2.4.6	helper_set_log_func	1073
9.2.4.7	helper_set_log_lvl	1073

9.2.4.8	libpack_GetVersion	1073
9.2.4.9	libpack_log	1073
9.2.4.10	unpack_result_code_only	1073
9.2.5	Variable Documentation	1073
9.2.5.1	glog	1073
9.2.5.2	gloglvl	1073
9.3	dms.h File Reference	1073
9.3.1	Macro Definition Documentation	1077
9.3.1.1	DMS_IMGDETAILS_LEN	1077
9.3.1.2	DMS_MAX_CUST_ID_LEN	1077
9.3.1.3	DMS_MAX_CUST_VALUE_LEN	1078
9.3.1.4	DMS_MAX_FWUPDATE_LOG_STR_SZ	1078
9.3.1.5	DMS_MAX_FWUPDATE_REF_STR_SZ	1078
9.3.1.6	DMS_PM_FACTORY	1078
9.3.1.7	DMS_PM_LOW	1078
9.3.1.8	DMS_PM_OFFLINE	1078
9.3.1.9	DMS_PM_ONLINE	1078
9.3.1.10	DMS_PM_PERSISTENT_LOW	1078
9.3.1.11	DMS_PM_RESET	1078
9.3.1.12	DMS_PM_SHUT_DOWN	1078
9.3.1.13	DMS_SET_REPORT_DISABLE	1078
9.3.1.14	DMS_SET_REPORT_ENABLE	1078
9.3.1.15	DMS_SLQSFWINFO_APPVERSION_SZ	1078
9.3.1.16	DMS_SLQSFWINFO_BOOTVERSION_SZ	1078
9.3.1.17	DMS_SLQSFWINFO_CARRIER_SZ	1078
9.3.1.18	DMS_SLQSFWINFO_CUR_CARR_NAME	1078
9.3.1.19	DMS_SLQSFWINFO_CUR_CARR_REV	1078
9.3.1.20	DMS_SLQSFWINFO_MODELID_SZ	1078
9.3.1.21	DMS_SLQSFWINFO_PACKAGEID_SZ	1078
9.3.1.22	DMS_SLQSFWINFO_PRIVERSION_SZ	1078
9.3.1.23	DMS_SLQSFWINFO_SKU_SZ	1078
9.3.1.24	DMS_SWI_SET_IND_DISABLE	1078
9.3.1.25	DMS_SWI_SET_IND_ENABLE	1078
9.3.1.26	DMS_UINT8_MAX_STRING_SZ	1078
9.3.1.27	MAX_BUILD_ID_LEN	1078
9.3.1.28	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	1078
9.3.1.29	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	1078
9.3.1.30	UNIQUE_ID_LEN	1078
9.3.2	Function Documentation	1079
9.3.2.1	pack_dms_GetActivationState	1079

9.3.2.2	<a href="#">pack_dms_GetBandCapability</a>	1079
9.3.2.3	<a href="#">pack_dms_GetCrashAction</a>	1079
9.3.2.4	<a href="#">pack_dms_GetCustFeature</a>	1080
9.3.2.5	<a href="#">pack_dms_GetCustFeaturesV2</a>	1080
9.3.2.6	<a href="#">pack_dms_GetDeviceCap</a>	1080
9.3.2.7	<a href="#">pack_dms_GetDeviceCapabilities</a>	1080
9.3.2.8	<a href="#">pack_dms_GetDeviceHardwareRev</a>	1081
9.3.2.9	<a href="#">pack_dms_GetDeviceMfr</a>	1081
9.3.2.10	<a href="#">pack_dms_GetDeviceSerialNumbers</a>	1081
9.3.2.11	<a href="#">pack_dms_GetFirmwareInfo</a>	1082
9.3.2.12	<a href="#">pack_dms_GetFirmwareRevision</a>	1082
9.3.2.13	<a href="#">pack_dms_GetFirmwareRevisions</a>	1082
9.3.2.14	<a href="#">pack_dms_GetFSN</a>	1083
9.3.2.15	<a href="#">pack_dms_GetHardwareRevision</a>	1083
9.3.2.16	<a href="#">pack_dms_GetIMSI</a>	1083
9.3.2.17	<a href="#">pack_dms_GetModelID</a>	1084
9.3.2.18	<a href="#">pack_dms_GetNetworkTime</a>	1084
9.3.2.19	<a href="#">pack_dms_GetPower</a>	1085
9.3.2.20	<a href="#">pack_dms_GetPRLVersion</a>	1085
9.3.2.21	<a href="#">pack_dms_GetSerialNumbers</a>	1085
9.3.2.22	<a href="#">pack_dms_GetUSBComp</a>	1086
9.3.2.23	<a href="#">pack_dms_GetVoiceNumber</a>	1086
9.3.2.24	<a href="#">pack_dms_SetCrashAction</a>	1086
9.3.2.25	<a href="#">pack_dms_SetCustFeature</a>	1087
9.3.2.26	<a href="#">pack_dms_SetCustFeaturesV2</a>	1087
9.3.2.27	<a href="#">pack_dms_SetEventReport</a>	1088
9.3.2.28	<a href="#">pack_dms_SetFirmwarePreference</a>	1088
9.3.2.29	<a href="#">pack_dms_SetPower</a>	1088
9.3.2.30	<a href="#">pack_dms_SetUSBComp</a>	1089
9.3.2.31	<a href="#">pack_dms_SLQSDmsSwiGetResetInfo</a>	1089
9.3.2.32	<a href="#">pack_dms_SLQSDmsSwiIndicationRegister</a>	1090
9.3.2.33	<a href="#">pack_dms_SLQSSwiGetBandCapability</a>	1090
9.3.2.34	<a href="#">pack_dms_SLQSSwiClearDyingGaspStatistics</a>	1090
9.3.2.35	<a href="#">pack_dms_SLQSSwiGetDyingGaspCfg</a>	1091
9.3.2.36	<a href="#">pack_dms_SLQSSwiGetDyingGaspStatistics</a>	1091
9.3.2.37	<a href="#">pack_dms_SLQSSwiGetFirmwareCurr</a>	1091
9.3.2.38	<a href="#">pack_dms_SLQSSwiGetFwUpdateStatus</a>	1092
9.3.2.39	<a href="#">pack_dms_SLQSSwiSetDyingGaspCfg</a>	1092
9.3.2.40	<a href="#">pack_dms_UIMGetICCID</a>	1092
9.3.2.41	<a href="#">unpack_dms_GetActivationState</a>	1093

9.3.2.42	<a href="#">unpack_dms_GetBandCapability</a>	1093
9.3.2.43	<a href="#">unpack_dms_GetCrashAction</a>	1094
9.3.2.44	<a href="#">unpack_dms_GetCustFeature</a>	1094
9.3.2.45	<a href="#">unpack_dms_GetCustFeaturesV2</a>	1094
9.3.2.46	<a href="#">unpack_dms_GetDeviceCap</a>	1094
9.3.2.47	<a href="#">unpack_dms_GetDeviceCapabilities</a>	1095
9.3.2.48	<a href="#">unpack_dms_GetDeviceHardwareRev</a>	1095
9.3.2.49	<a href="#">unpack_dms_GetDeviceMfr</a>	1095
9.3.2.50	<a href="#">unpack_dms_GetDeviceSerialNumbers</a>	1096
9.3.2.51	<a href="#">unpack_dms_GetFirmwareInfo</a>	1096
9.3.2.52	<a href="#">unpack_dms_GetFirmwareRevision</a>	1096
9.3.2.53	<a href="#">unpack_dms_GetFirmwareRevisions</a>	1097
9.3.2.54	<a href="#">unpack_dms_GetFSN</a>	1097
9.3.2.55	<a href="#">unpack_dms_GetHardwareRevision</a>	1097
9.3.2.56	<a href="#">unpack_dms_GetIMSI</a>	1098
9.3.2.57	<a href="#">unpack_dms_GetModelID</a>	1098
9.3.2.58	<a href="#">unpack_dms_GetNetworkTime</a>	1098
9.3.2.59	<a href="#">unpack_dms_GetPower</a>	1099
9.3.2.60	<a href="#">unpack_dms_GetPRLVersion</a>	1099
9.3.2.61	<a href="#">unpack_dms_GetSerialNumbers</a>	1099
9.3.2.62	<a href="#">unpack_dms_GetUSBComp</a>	1100
9.3.2.63	<a href="#">unpack_dms_GetVoiceNumber</a>	1100
9.3.2.64	<a href="#">unpack_dms_SetCrashAction</a>	1100
9.3.2.65	<a href="#">unpack_dms_SetCustFeature</a>	1101
9.3.2.66	<a href="#">unpack_dms_SetCustFeaturesV2</a>	1101
9.3.2.67	<a href="#">unpack_dms_SetEventReport</a>	1101
9.3.2.68	<a href="#">unpack_dms_SetEventReport_ind</a>	1102
9.3.2.69	<a href="#">unpack_dms_SetFirmwarePreference</a>	1102
9.3.2.70	<a href="#">unpack_dms_SetPower</a>	1102
9.3.2.71	<a href="#">unpack_dms_SetUSBComp</a>	1103
9.3.2.72	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo</a>	1103
9.3.2.73	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_Ind</a>	1103
9.3.2.74	<a href="#">unpack_dms_SLQSDmsSwiIndicationRegister</a>	1104
9.3.2.75	<a href="#">unpack_dms_SLQSGetBandCapability</a>	1104
9.3.2.76	<a href="#">unpack_dms_SLQSSwiClearDyingGaspStatistics</a>	1105
9.3.2.77	<a href="#">unpack_dms_SLQSSwiGetDyingGaspCfg</a>	1105
9.3.2.78	<a href="#">unpack_dms_SLQSSwiGetDyingGaspStatistics</a>	1105
9.3.2.79	<a href="#">unpack_dms_SLQSSwiGetFirmwareCurr</a>	1106
9.3.2.80	<a href="#">unpack_dms_SLQSSwiGetFwUpdateStatus</a>	1106
9.3.2.81	<a href="#">unpack_dms_SLQSSwiSetDyingGaspCfg</a>	1106

9.3.2.82	unpack_dms_UIMGetICCID	1107
9.4	fms.h File Reference	1107
9.4.1	Macro Definition Documentation	1108
9.4.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	1108
9.4.1.2	FMS_GOBI_LISTENTRIES_MAX	1108
9.4.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	1108
9.4.1.4	FMS_GOBI_MBN_IMG_ID_STR_LEN	1108
9.4.1.5	FMS_IMAGE_ID_BUILD_ID_LEN	1108
9.4.1.6	FMS_IMAGE_ID_IMG_ID_LEN	1108
9.4.1.7	FMS_IMAGE_ID_MAX_ENTRIES	1108
9.4.1.8	FMS_IMAGE_ID_PRI_IMGTYPE	1108
9.4.1.9	FMS_MAX_IMAGE_ID_ELEMENT	1108
9.4.1.10	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	1108
9.4.2	Function Documentation	1108
9.4.2.1	GetValidFwPriCombinations	1108
9.4.2.2	pack_fms_GetImagesPreference	1109
9.4.2.3	pack_fms_GetStoredImages	1109
9.4.2.4	pack_fms_SetImagesPreference	1109
9.4.2.5	unpack_fms_GetImagesPreference	1110
9.4.2.6	unpack_fms_GetStoredImages	1110
9.4.2.7	unpack_fms_SetImagesPreference	1110
9.5	loc.h File Reference	1110
9.5.1	Macro Definition Documentation	1112
9.5.1.1	LOC_UINT8_MAX_STRING_SZ	1112
9.5.1.2	LOCEVENTMASKBATCHFULLNOTIFICATION	1112
9.5.1.3	LOCEVENTMASKENGINESTATE	1112
9.5.1.4	LOCEVENTMASKFIXSESSIONSTATE	1112
9.5.1.5	LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION	1112
9.5.1.6	LOCEVENTMASKGEOFENCEBREACHNOTIFICATION	1113
9.5.1.7	LOCEVENTMASKGEOFENCEGENALERT	1113
9.5.1.8	LOCEVENTMASKGNSSMEASUREMENTREPORT	1113
9.5.1.9	LOCEVENTMASKGNSSSVINFO	1113
9.5.1.10	LOCEVENTMASKINJECTPOSITIONREQ	1113
9.5.1.11	LOCEVENTMASKINJECTPREDICTEDORBITSREQ	1113
9.5.1.12	LOCEVENTMASKINJECTTIMEREQ	1113
9.5.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	1113
9.5.1.14	LOCEVENTMASKINVALIDVALUE	1113
9.5.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	1113
9.5.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	1113
9.5.1.17	LOCEVENTMASKMOTIONDATACONTROL	1114

9.5.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	1114
9.5.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	1114
9.5.1.20	LOCEVENTMASKNMEA	1114
9.5.1.21	LOCEVENTMASKPEDOMETERCONTROL	1114
9.5.1.22	LOCEVENTMASKPOSITIONREPORT	1114
9.5.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	1114
9.5.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	1114
9.5.1.25	LOCEVENTMASKTIMESYNCREQ	1114
9.5.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	1114
9.5.1.27	LOCEVENTMASKWIFIREQ	1114
9.5.2	Enumeration Type Documentation	1115
9.5.2.1	anonymous enum	1115
9.5.3	Function Documentation	1115
9.5.3.1	pack_loc_DeleteAssistData	1115
9.5.3.2	pack_loc_EventRegister	1115
9.5.3.3	pack_loc_SetExtPowerState	1115
9.5.3.4	pack_loc_SetOperationMode	1116
9.5.3.5	pack_loc_SLQSLOCGetBestAvailPos	1116
9.5.3.6	pack_loc_Start	1116
9.5.3.7	pack_loc_Stop	1117
9.5.3.8	unpack_loc_BestAvailPos_Ind	1117
9.5.3.9	unpack_loc_DeleteAssistData	1117
9.5.3.10	unpack_loc_EngineState_Ind	1118
9.5.3.11	unpack_loc_EventRegister	1118
9.5.3.12	unpack_loc_PositionRpt_Ind	1118
9.5.3.13	unpack_loc_SetExtPowerConfig_Ind	1119
9.5.3.14	unpack_loc_SetExtPowerState	1119
9.5.3.15	unpack_loc_SetOperationMode	1120
9.5.3.16	unpack_loc_SLQSLOCGetBestAvailPos	1120
9.5.3.17	unpack_loc_Start	1120
9.5.3.18	unpack_loc_Stop	1121
9.6	nas.h File Reference	1121
9.6.1	Macro Definition Documentation	1126
9.6.1.1	NAS_MAX_DESCRIPTION_LENGTH	1126
9.6.1.2	NAS_MAX_NUM_NETWORKS	1126
9.6.1.3	NAS_OTA_MESSAGE_MAX_BUF_SIZE	1126
9.6.1.4	NAS_PLMN_LENGTH	1127
9.6.1.5	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	1127
9.6.2	Enumeration Type Documentation	1127
9.6.2.1	LIBPACK_NAS_LTE_CPHY_CA_BW_NRB	1127

9.6.2.2	LIBPACK_NAS_LTE_CPHY_SCELL_STATE . . . . .	1127
9.6.2.3	NAS_LTE_CPHY_CA_BW_NRB_LITE . . . . .	1127
9.6.2.4	NAS_LTE_CPHY_SCELL_STATE_LITE . . . . .	1127
9.6.3	Function Documentation . . . . .	1127
9.6.3.1	pack_nas_GetACCOLC . . . . .	1127
9.6.3.2	pack_nas_GetANAAAAAuthenticationStatus . . . . .	1128
9.6.3.3	pack_nas_GetCDMANetworkParameters . . . . .	1128
9.6.3.4	pack_nas_GetHomeNetwork . . . . .	1128
9.6.3.5	pack_nas_GetNetworkPreference . . . . .	1129
9.6.3.6	pack_nas_GetRFInfo . . . . .	1129
9.6.3.7	pack_nas_GetServingNetwork . . . . .	1129
9.6.3.8	pack_nas_GetServingNetworkCapabilities . . . . .	1129
9.6.3.9	pack_nas_GetSignalStrengths . . . . .	1130
9.6.3.10	pack_nas_PerformNetworkScan . . . . .	1130
9.6.3.11	pack_nas_SetACCOLC . . . . .	1130
9.6.3.12	pack_nas_SetLURejectCallback . . . . .	1131
9.6.3.13	pack_nas_SetNetworkPreference . . . . .	1131
9.6.3.14	pack_nas_SetRFInfoCallback . . . . .	1131
9.6.3.15	pack_nas_SlqsGetLTECphyCAInfo . . . . .	1131
9.6.3.16	pack_nas_SLQSGetNetworkTime . . . . .	1132
9.6.3.17	pack_nas_SLQSGetPLMNName . . . . .	1132
9.6.3.18	pack_nas_SLQSGetServingSystem . . . . .	1132
9.6.3.19	pack_nas_SLQSGetSignalStrength . . . . .	1133
9.6.3.20	pack_nas_SLQSGetSysInfo . . . . .	1133
9.6.3.21	pack_nas_SLQSGetSysSelectionPref . . . . .	1133
9.6.3.22	pack_nas_SLQSInitiateNetworkRegistration . . . . .	1133
9.6.3.23	pack_nas_SLQSNasConfigSigInfo2 . . . . .	1134
9.6.3.24	pack_nas_SLQSNasGetCellLocationInfo . . . . .	1134
9.6.3.25	pack_nas_SLQSNasGetSigInfo . . . . .	1134
9.6.3.26	pack_nas_SLQSNasIndicationRegisterExt . . . . .	1135
9.6.3.27	pack_nas_SLQSNasSwiModemStatus . . . . .	1135
9.6.3.28	pack_nas_SLQSNasSwiOTAMessageCallback . . . . .	1135
9.6.3.29	pack_nas_SLQSSetBandPreference . . . . .	1136
9.6.3.30	pack_nas_SLQSSetSignalStrengthsCallback . . . . .	1136
9.6.3.31	pack_nas_SLQSSetSysSelectionPref . . . . .	1136
9.6.3.32	pack_nas_SLQSSwiGetLteCQI . . . . .	1137
9.6.3.33	unpack_nas_GetACCOLC . . . . .	1137
9.6.3.34	unpack_nas_GetANAAAAAuthenticationStatus . . . . .	1137
9.6.3.35	unpack_nas_GetCDMANetworkParameters . . . . .	1138
9.6.3.36	unpack_nas_GetHomeNetwork . . . . .	1138

9.6.3.37	<a href="#">unpack_nas_GetNetworkPreference</a>	1138
9.6.3.38	<a href="#">unpack_nas_GetRFInfo</a>	1138
9.6.3.39	<a href="#">unpack_nas_GetServingNetwork</a>	1139
9.6.3.40	<a href="#">unpack_nas_GetServingNetworkCapabilities</a>	1139
9.6.3.41	<a href="#">unpack_nas_GetSignalStrengths</a>	1139
9.6.3.42	<a href="#">unpack_nas_PerformNetworkScan</a>	1140
9.6.3.43	<a href="#">unpack_nas_SetACCOLC</a>	1140
9.6.3.44	<a href="#">unpack_nas_SetDataCapabilitiesCallback_ind</a>	1140
9.6.3.45	<a href="#">unpack_nas_SetEventReportInd</a>	1141
9.6.3.46	<a href="#">unpack_nas_SetLURejectCallback</a>	1141
9.6.3.47	<a href="#">unpack_nas_SetNasLTECphyCaIndCallback_ind</a>	1141
9.6.3.48	<a href="#">unpack_nas_SetNetworkPreference</a>	1141
9.6.3.49	<a href="#">unpack_nas_SetRFInfoCallback</a>	1141
9.6.3.50	<a href="#">unpack_nas_SetRoamingIndicatorCallback_ind</a>	1142
9.6.3.51	<a href="#">unpack_nas_SetServingSystemCallback_ind</a>	1142
9.6.3.52	<a href="#">unpack_nas_SLqsGetLTECphyCAInfo</a>	1142
9.6.3.53	<a href="#">unpack_nas_SLQSGetNetworkTime</a>	1142
9.6.3.54	<a href="#">unpack_nas_SLQSGetPLMNName</a>	1142
9.6.3.55	<a href="#">unpack_nas_SLQSGetServingSystem</a>	1143
9.6.3.56	<a href="#">unpack_nas_SLQSGetSignalStrength</a>	1143
9.6.3.57	<a href="#">unpack_nas_SLQSGetSysInfo</a>	1143
9.6.3.58	<a href="#">unpack_nas_SLQSGetSysSelectionPref</a>	1144
9.6.3.59	<a href="#">unpack_nas_SLQSInitiateNetworkRegistration</a>	1144
9.6.3.60	<a href="#">unpack_nas_SLQSNasConfigSigInfo2</a>	1144
9.6.3.61	<a href="#">unpack_nas_SLQSNasGetCellLocationInfo</a>	1144
9.6.3.62	<a href="#">unpack_nas_SLQSNasGetSigInfo</a>	1145
9.6.3.63	<a href="#">unpack_nas_SLQSNasIndicationRegisterExt</a>	1145
9.6.3.64	<a href="#">unpack_nas_SLQSNasNetworkTimeCallBack_ind</a>	1145
9.6.3.65	<a href="#">unpack_nas_SLQSNasSigInfoCallback_ind</a>	1146
9.6.3.66	<a href="#">unpack_nas_SLQSNasSwiModemStatus</a>	1146
9.6.3.67	<a href="#">unpack_nas_SLQSNasSwiOTAMessageCallback</a>	1146
9.6.3.68	<a href="#">unpack_nas_SLQSNasSwiOTAMessageCallback_ind</a>	1147
9.6.3.69	<a href="#">unpack_nas_SLQSNasSysInfoCallback_ind</a>	1147
9.6.3.70	<a href="#">unpack_nas_SLQSSetBandPreference</a>	1147
9.6.3.71	<a href="#">unpack_nas_SLQSSetSignalStrengthsCallback</a>	1147
9.6.3.72	<a href="#">unpack_nas_SLQSSetSysSelectionPref</a>	1148
9.6.3.73	<a href="#">unpack_nas_SLQSSetSysSelectionPrefCallBack_ind</a>	1148
9.6.3.74	<a href="#">unpack_nas_SLQSSwiGetLteCQI</a>	1148
9.7	<a href="#">qaCbkCatEventReportInd.h File Reference</a>	1149
9.7.1	<a href="#">Macro Definition Documentation</a>	1149

9.7.1.1	QMI_CAN_COMMON_EVENT_TLV_NUMBER . . . . .	1149
9.7.1.2	QMI_MAX_CAT_EVENT_DATA_LENGTH . . . . .	1149
9.7.2	Enumeration Type Documentation . . . . .	1150
9.7.2.1	eQMI_CAT_EVENT_REPORT_IND_TLV . . . . .	1150
9.7.2.2	eQMI_CAT_EVENT_REPORT_IND_TLV_LENGTH . . . . .	1150
9.7.3	Function Documentation . . . . .	1150
9.7.3.1	UpkQmiCbkCatEventReportInd . . . . .	1150
9.8	qaCbkSwiOmaDmEventReportInd.h File Reference . . . . .	1150
9.8.1	Macro Definition Documentation . . . . .	1151
9.8.1.1	QMI_SWIOMA_DM_CONFIG . . . . .	1151
9.8.1.2	QMI_SWIOMA_DM_FOTA . . . . .	1151
9.8.1.3	QMI_SWIOMA_DM_NOT . . . . .	1151
9.8.2	Enumeration Type Documentation . . . . .	1151
9.8.2.1	eQMI_SWIOMA_DM_EVENT_REPORT_IND . . . . .	1151
9.8.3	Function Documentation . . . . .	1151
9.8.3.1	UpkQmiCbkSwiOmaDmEventReportInd . . . . .	1151
9.8.3.2	UpkQmiCbkSwiOmaDmEventReportIndExt . . . . .	1151
9.9	qaGobiApiAudio.h File Reference . . . . .	1151
9.9.1	Detailed Description . . . . .	1152
9.9.2	Function Documentation . . . . .	1152
9.9.2.1	SLQSGetAudioPathConfig . . . . .	1152
9.9.2.2	SLQSGetAudioProfile . . . . .	1152
9.9.2.3	SLQSGetAudioVolTLBConfig . . . . .	1153
9.9.2.4	SLQSSetAudioPathConfig . . . . .	1153
9.9.2.5	SLQSSetAudioProfile . . . . .	1154
9.9.2.6	SLQSSetAudioVolTLBConfig . . . . .	1154
9.10	qaGobiApiCat.h File Reference . . . . .	1155
9.10.1	Detailed Description . . . . .	1155
9.10.2	Function Documentation . . . . .	1155
9.10.2.1	CATSendEnvelopeCommand . . . . .	1155
9.10.2.2	CATSendTerminalResponse . . . . .	1156
9.11	qaGobiApiCbk.h File Reference . . . . .	1156
9.11.1	Detailed Description . . . . .	1164
9.11.2	Macro Definition Documentation . . . . .	1164
9.11.2.1	CBK_DISABLE_EVENT . . . . .	1164
9.11.2.2	CBK_ENABLE_EVENT . . . . .	1164
9.11.2.3	CBK_NOCHANGE . . . . .	1164
9.11.2.4	DEREGISTER_EVENT . . . . .	1164
9.11.2.5	DEREGISTER_SRV . . . . .	1164
9.11.2.6	DHCP_MAX_NUM_OPTIONS . . . . .	1164

9.11.2.7 DHCP_OPTION_DATA_BUF_SIZE . . . . .	1164
9.11.2.8 EVENT_MASK_CARD . . . . .	1164
9.11.2.9 EVENT_MASK_DEREGISTER_ALL . . . . .	1164
9.11.2.10 EVENT_MASK_PHY_SLOT_STATUS . . . . .	1164
9.11.2.11 FIRST_INSTANCE . . . . .	1164
9.11.2.12 INVALID_INSTACNE . . . . .	1164
9.11.2.13 IPV4 . . . . .	1164
9.11.2.14 IPV4V6 . . . . .	1164
9.11.2.15 IPV6 . . . . .	1164
9.11.2.16 LOC_EVENT_MASK_ENG_STATE . . . . .	1164
9.11.2.17 LOC_EVENT_MASK_GNSS_SV_INFO . . . . .	1165
9.11.2.18 LOC_EVENT_MASK_INJECT_TIME . . . . .	1165
9.11.2.19 LOC_EVENT_MASK_SENSOR_STREAM . . . . .	1165
9.11.2.20 LOC_EVENT_MASK_TIME_SYNC . . . . .	1165
9.11.2.21 LOC_EVENT_POSITION_REPORT . . . . .	1165
9.11.2.22 MAX_MITIGATION_DEV_ID_LEN . . . . .	1165
9.11.2.23 MAX_NO_OF_APPLICATIONS . . . . .	1165
9.11.2.24 MAX_NO_OF_CALLS . . . . .	1165
9.11.2.25 MAX_NO_OF_FILES . . . . .	1165
9.11.2.26 MAX_NO_OF_SLOTS . . . . .	1165
9.11.2.27 MAX_NO_OF_UUSINFO . . . . .	1165
9.11.2.28 MAX_PATH_LENGTH . . . . .	1165
9.11.2.29 MAX_RADIO_INTERFACE_LIST . . . . .	1165
9.11.2.30 MAXUSSDLENGTH . . . . .	1165
9.11.2.31 NAS_SRV . . . . .	1165
9.11.2.32 NUM_OF_SET . . . . .	1165
9.11.2.33 PDS_SRV . . . . .	1165
9.11.2.34 QMI_ETWS_MAX_PAYLOAD_LENGTH . . . . .	1165
9.11.2.35 QMI_MAX_VOICE_NUMBER_LENGTH . . . . .	1165
9.11.2.36 QMI_WMS_MAX_PAYLOAD_LENGTH . . . . .	1165
9.11.2.37 REGISTER_EVENT . . . . .	1165
9.11.2.38 REGISTER_SRV . . . . .	1165
9.11.2.39 SECOND_INSTANCE . . . . .	1165
9.11.2.40 SIGSTRENGTH_THRESHOLD_ARR_SZ . . . . .	1165
9.11.2.41 THIRD_INSTANCE . . . . .	1165
9.11.2.42 USSD_DCS_8BIT . . . . .	1165
9.11.2.43 USSD_DCS_ASCII . . . . .	1165
9.11.2.44 USSD_DCS_UCS2 . . . . .	1165
9.11.2.45 VOICE_SRV . . . . .	1166
9.11.2.46 WDS_SRV . . . . .	1166

9.11.3 Typedef Documentation . . . . .	1166
9.11.3.1 accelAcceptReady . . . . .	1166
9.11.3.2 accelTempAcceptReady . . . . .	1166
9.11.3.3 eDevState . . . . .	1167
9.11.3.4 eSMSEventType . . . . .	1167
9.11.3.5 gpsTime . . . . .	1167
9.11.3.6 gyroAcceptReady . . . . .	1167
9.11.3.7 gyroTempAcceptReady . . . . .	1168
9.11.3.8 LteNasReleaseInfo . . . . .	1168
9.11.3.9 modemTempNotification . . . . .	1168
9.11.3.10 packetSrvStatus . . . . .	1169
9.11.3.11 precisionDilution . . . . .	1170
9.11.3.12 ResetInfoNotification . . . . .	1170
9.11.3.13 sensorDataUsage . . . . .	1171
9.11.3.14 sessionInformation . . . . .	1172
9.11.3.15 sessionInformationExt . . . . .	1172
9.11.3.16 SMSAsyncRawSend . . . . .	1172
9.11.3.17 SMSCAddressInfo . . . . .	1173
9.11.3.18 SMSEtwsMessageInfo . . . . .	1173
9.11.3.19 SMSEtwsPlmnInfo . . . . .	1173
9.11.3.20 SMSEventInfo . . . . .	1174
9.11.3.21 SMSMessageModelInfo . . . . .	1174
9.11.3.22 SMSMTMessageInfo . . . . .	1174
9.11.3.23 SMSOnIMSInfo . . . . .	1175
9.11.3.24 SMSTransferRouteMTMessageInfo . . . . .	1175
9.11.3.25 svUsedforFix . . . . .	1175
9.11.3.26 SwiOTAMsg . . . . .	1176
9.11.3.27 tFNActivationStatus . . . . .	1176
9.11.3.28 tFNAllCallStatus . . . . .	1177
9.11.3.29 tFNASwiLTECphyCallInfo . . . . .	1177
9.11.3.30 tFNASwiOTAMsg . . . . .	1177
9.11.3.31 tFNAsyncRawSend . . . . .	1177
9.11.3.32 tFNBandPreference . . . . .	1178
9.11.3.33 tFNBestAvailPos . . . . .	1180
9.11.3.34 tFNCATEvent . . . . .	1180
9.11.3.35 tFNCbkUimSlotStatusChangeInd . . . . .	1180
9.11.3.36 tFNDataCapabilities . . . . .	1180
9.11.3.37 tFNDataSysStatus . . . . .	1181
9.11.3.38 tFNDeIAssistData . . . . .	1181
9.11.3.39 tFNDeviceStateChange . . . . .	1181

9.11.3.40 tFNDHCPv4ClientLeaseStatus . . . . .	1182
9.11.3.41 tFNDTMFEvent . . . . .	1182
9.11.3.42 tFNDUNCallInfo . . . . .	1182
9.11.3.43 tFNEventPosition . . . . .	1182
9.11.3.44 tFNFwDIdCompletion . . . . .	1182
9.11.3.45 tFNGnssSvInfo . . . . .	1183
9.11.3.46 tFNHDRPersonality . . . . .	1183
9.11.3.47 tFNImsaPdpStatus . . . . .	1183
9.11.3.48 tFNImsaRatStatus . . . . .	1183
9.11.3.49 tFNImsaRegStatus . . . . .	1183
9.11.3.50 tFNImsaSvcStatus . . . . .	1184
9.11.3.51 tFNImRegMgrConfig . . . . .	1184
9.11.3.52 tFNImSIPConfig . . . . .	1184
9.11.3.53 tFNImSMSConfig . . . . .	1184
9.11.3.54 tFNImUserConfig . . . . .	1184
9.11.3.55 tFNImVoIPConfig . . . . .	1185
9.11.3.56 tFNInfoRec . . . . .	1185
9.11.3.57 tFNInjectPosition . . . . .	1185
9.11.3.58 tFNInjectSensorData . . . . .	1185
9.11.3.59 tFNInjectTimeStatus . . . . .	1185
9.11.3.60 tFNInjectUTCTime . . . . .	1185
9.11.3.61 tFNLURreject . . . . .	1186
9.11.3.62 tFNMemoryFull . . . . .	1187
9.11.3.63 tFNMessageWaiting . . . . .	1187
9.11.3.64 tFNMitiLvlRpt . . . . .	1187
9.11.3.65 tFNMobileIPStatus . . . . .	1187
9.11.3.66 tFNModemTempInfo . . . . .	1187
9.11.3.67 tFNNet . . . . .	1188
9.11.3.68 tFNNetworkTime . . . . .	1188
9.11.3.69 tFNNewGPS . . . . .	1188
9.11.3.70 tFNNewNMEA . . . . .	1189
9.11.3.71 tFNNewRMTransferStatistics . . . . .	1189
9.11.3.72 tFNNewSMS . . . . .	1190
9.11.3.73 tFNOMADMState . . . . .	1190
9.11.3.74 tFNOpMode . . . . .	1191
9.11.3.75 tFNOTASPStatus . . . . .	1191
9.11.3.76 tFNPacketSrvState . . . . .	1191
9.11.3.77 tFNPDSSState . . . . .	1191
9.11.3.78 tFNPower . . . . .	1191
9.11.3.79 tFNPrivacyChange . . . . .	1192

9.11.3.80	tFNQosNWStatus	1192
9.11.3.81	tFNQosPriEvent	1192
9.11.3.82	tFNQosStatus	1192
9.11.3.83	tFNRankIndicator	1193
9.11.3.84	tFNResetInfo	1193
9.11.3.85	tFNRFInfo	1194
9.11.3.86	tFNRoamingIndicator	1194
9.11.3.87	tFNSDKTerminated	1194
9.11.3.88	tFNSensorStreaming	1195
9.11.3.89	tFNServingSystem	1195
9.11.3.90	tFNSetCradleMount	1195
9.11.3.91	tFNSetEngineState	1195
9.11.3.92	tFNSetEventTimeSync	1195
9.11.3.93	tFNSetExtPowerConfig	1195
9.11.3.94	tFNSigInfo	1195
9.11.3.95	tFNSignalStrength	1195
9.11.3.96	tFNSLQSOMADMAAlert	1195
9.11.3.97	tFNSLQSQOSEvent	1196
9.11.3.98	tFNSLQSSessionState	1196
9.11.3.99	tFNSLQSSignalStrengths	1196
9.11.3.100	tFNSLQSWDSEvent	1196
9.11.3.101	tFNSMSEvents	1197
9.11.3.102	tFNSUPSInfo	1197
9.11.3.103	tFNSUPSNotification	1197
9.11.3.104	tFNSysInfo	1197
9.11.3.105	tFNSysSelectionPref	1197
9.11.3.106	tFNtransLayerInfo	1198
9.11.3.107	tFNtransNWRegInfo	1198
9.11.3.108	tFNUIMRefresh	1198
9.11.3.109	tFNUIMStatusChangeInfo	1198
9.11.3.110	tFNUSSDNotification	1198
9.11.3.111	tFNUSSDNoWaitIndication	1199
9.11.3.112	tFNUSSDRelease	1199
9.11.3.113	transLayerNotification	1199
9.11.3.114	transNWRegInfoNotification	1199
9.11.4	Enumeration Type Documentation	1200
9.11.4.1	device_state_enum	1200
9.11.4.2	eQaQMIService	1200
9.11.4.3	SMSEventType	1200
9.11.5	Function Documentation	1201

9.11.5.1	iSetCATEventCallback . . . . .	1201
9.11.5.2	iSetSignalStrengthCallback . . . . .	1201
9.11.5.3	iSLQSSetDUNCallInfoCallback . . . . .	1201
9.11.5.4	iSLQSSetSignalStrengthsCallback . . . . .	1201
9.11.5.5	iSLQSSetWdsFirstInstEventCallback . . . . .	1201
9.11.5.6	iSLQSSetWdsSecondInstEventCallback . . . . .	1201
9.11.5.7	iSLQSSetWdsThirdInstEventCallback . . . . .	1201
9.11.5.8	iSLQSSetWdsXferStatsFirstInstCallback . . . . .	1201
9.11.5.9	iSLQSSetWdsXferStatsSecondInstCallback . . . . .	1201
9.11.5.10	SetActivationStatusCallback . . . . .	1201
9.11.5.11	SetCATEventCallback . . . . .	1201
9.11.5.12	SetDataCapabilitiesCallback . . . . .	1202
9.11.5.13	SetDeviceStateChangeCbk . . . . .	1203
9.11.5.14	SetFwDIdCompletionCbk . . . . .	1203
9.11.5.15	SetGPSCallback . . . . .	1204
9.11.5.16	SetLocBestAvailPosCallback . . . . .	1204
9.11.5.17	SetLocCradleMountCallback . . . . .	1204
9.11.5.18	SetLocDeleteAssistDataCallback . . . . .	1205
9.11.5.19	SetLocEngineStateCallback . . . . .	1205
9.11.5.20	SetLocEventPositionCallback . . . . .	1205
9.11.5.21	SetLocEventTimeSyncCallback . . . . .	1205
9.11.5.22	SetLocGnssSvInfoCallback . . . . .	1206
9.11.5.23	SetLocInjectSensorDataCallback . . . . .	1206
9.11.5.24	SetLocInjectTimeCallback . . . . .	1206
9.11.5.25	SetLocOpModeCallback . . . . .	1207
9.11.5.26	SetLocSensorStreamingCallback . . . . .	1207
9.11.5.27	SetLocSetExtPowerConfigCallback . . . . .	1207
9.11.5.28	SetLURejectCallback . . . . .	1207
9.11.5.29	SetMobileIPStatusCallback . . . . .	1208
9.11.5.30	SetNasLTECphyCaIndCallback . . . . .	1208
9.11.5.31	SetNetChangeCbk . . . . .	1209
9.11.5.32	SetNewSMSCallback . . . . .	1209
9.11.5.33	SetNMEACallback . . . . .	1210
9.11.5.34	SetOMADMStateCallback . . . . .	1210
9.11.5.35	SetPDSSStateCallback . . . . .	1210
9.11.5.36	SetPowerCallback . . . . .	1211
9.11.5.37	SetRankIndicatorCallback . . . . .	1211
9.11.5.38	SetRFInfoCallback . . . . .	1211
9.11.5.39	SetRMTransferStatisticsCallback . . . . .	1212
9.11.5.40	SetRoamingIndicatorCallback . . . . .	1212

9.11.5.41 SetSignalStrengthCallback . . . . .	1212
9.11.5.42 SetSLQSOMADMAAlertCallback . . . . .	1213
9.11.5.43 SetSLQSOMADMAAlertCallbackExt . . . . .	1213
9.11.5.44 SetUimSlotStatusChangeCallback . . . . .	1214
9.11.5.45 SetUSSDNotificationCallback . . . . .	1214
9.11.5.46 SetUSSDNoWaitIndicationCallback . . . . .	1214
9.11.5.47 SetUSSDReleaseCallback . . . . .	1215
9.11.5.48 SLQSNasNetworkTimeCallBack . . . . .	1215
9.11.5.49 SLQSNasSigInfo2CallBack . . . . .	1216
9.11.5.50 SLQSNasSigInfoCallBack . . . . .	1216
9.11.5.51 SLQSNasSwiOTAMessageCallback . . . . .	1217
9.11.5.52 SLQSNasSysInfoCallBack . . . . .	1217
9.11.5.53 SLQSSetBandPreferenceCbk . . . . .	1218
9.11.5.54 SLQSSetDataSystemStatusCallback . . . . .	1218
9.11.5.55 SLQSSetDHCIPv4ClientLeaseStatusCallback . . . . .	1218
9.11.5.56 SLQSSetDUNCallInfoCallback . . . . .	1219
9.11.5.57 SLQSSetIMSAPdpStatusCallback . . . . .	1219
9.11.5.58 SLQSSetIMSAratStatusCallback . . . . .	1220
9.11.5.59 SLQSSetIMSAREgStatusCallback . . . . .	1220
9.11.5.60 SLQSSetIMSASvcStatusCallback . . . . .	1220
9.11.5.61 SLQSSetIMSSMSConfigCallback . . . . .	1221
9.11.5.62 SLQSSetIMSUserConfigCallback . . . . .	1221
9.11.5.63 SLQSSetIMSVoIPConfigCallback . . . . .	1222
9.11.5.64 SLQSSetLocInjectPositionCallback . . . . .	1222
9.11.5.65 SLQSSetLocInjectUTCTimeCallback . . . . .	1222
9.11.5.66 SLQSSetModemTempCallback . . . . .	1223
9.11.5.67 SLQSSetPacketSrvStatusCallback . . . . .	1223
9.11.5.68 SLQSSetQosEventCallback . . . . .	1224
9.11.5.69 SLQSSetQosNWStatusCallback . . . . .	1224
9.11.5.70 SLQSSetQosPriEventCallback . . . . .	1224
9.11.5.71 SLQSSetQosStatusCallback . . . . .	1225
9.11.5.72 SLQSSetRegMgrConfigCallback . . . . .	1225
9.11.5.73 SLQSSetSDKTerminatedCallback . . . . .	1226
9.11.5.74 SLQSSetServingSystemCallback . . . . .	1226
9.11.5.75 SLQSSetSessionStateCallback . . . . .	1227
9.11.5.76 SLQSSetSignalStrengthsCallback . . . . .	1227
9.11.5.77 SLQSSetSIPConfigCallback . . . . .	1228
9.11.5.78 SLQSSetSMSEventCallback . . . . .	1228
9.11.5.79 SLQSSetSwiGetResetInfoCallback . . . . .	1228
9.11.5.80 SLQSSetSwiHDRPersCallback . . . . .	1229

9.11.5.81 SLQSSetSysSelectionPrefCallBack . . . . .	1229
9.11.5.82 SLQSSetTransLayerInfoCallBack . . . . .	1229
9.11.5.83 SLQSSetTransNWRegInfoCallBack . . . . .	1230
9.11.5.84 SLQSSetWdsEventCallBack . . . . .	1230
9.11.5.85 SLQSSetWdsTransferStatisticCallBack . . . . .	1231
9.11.5.86 SLQSTmdMitigationLvlRptCallBack . . . . .	1232
9.11.5.87 SLQSUIIMSetRefreshCallBack . . . . .	1232
9.11.5.88 SLQSUIIMSetStatusChangeCallBack . . . . .	1232
9.11.5.89 SLQSVoiceInfoRecCallBack . . . . .	1233
9.11.5.90 SLQSVoiceSetAllCallStatusCallBack . . . . .	1233
9.11.5.91 SLQSVoiceSetDTMFEventCallBack . . . . .	1234
9.11.5.92 SLQSVoiceSetOTASPStatusCallBack . . . . .	1234
9.11.5.93 SLQSVoiceSetPrivacyChangeCallBack . . . . .	1234
9.11.5.94 SLQSVoiceSetSUPSCallBack . . . . .	1235
9.11.5.95 SLQSVoiceSetSUPSNotificationCallBack . . . . .	1235
9.11.5.96 SLQSWmsAsyncRawSendCallBack . . . . .	1236
9.11.5.97 SLQSWmsMemoryFullCallBack . . . . .	1236
9.11.5.98 SLQSWmsMessageWaitingCallBack . . . . .	1237
9.12 qaGobiApiDcs.h File Reference . . . . .	1237
9.12.1 Detailed Description . . . . .	1238
9.12.2 Macro Definition Documentation . . . . .	1238
9.12.2.1 LEN . . . . .	1238
9.12.2.2 PORTNAM_LEN . . . . .	1238
9.12.3 Function Documentation . . . . .	1238
9.12.3.1 QCWWAN2kConnect . . . . .	1238
9.12.3.2 QCWWAN2kEnumerateDevices . . . . .	1239
9.12.3.3 QCWWAN2kGetConnectedDeviceID . . . . .	1239
9.12.3.4 QCWWANConnect . . . . .	1240
9.12.3.5 QCWWANDisconnect . . . . .	1240
9.12.3.6 QCWWANEnumerateDevices . . . . .	1240
9.12.3.7 SetSDKImagePath . . . . .	1241
9.12.3.8 SLQSGetDeviceMode . . . . .	1241
9.12.3.9 SLQSGetNetStatistic . . . . .	1241
9.12.3.10 SLQSGetUsbPortNames . . . . .	1242
9.12.3.11 SLQSKillSDKProcess . . . . .	1242
9.12.3.12 SLQSSetLoggingMask . . . . .	1243
9.12.3.13 SLQSStart . . . . .	1243
9.12.3.14 SLQSStart_AVAgent . . . . .	1244
9.12.3.15 SLQSStartSrv . . . . .	1244
9.13 qaGobiApiDms.h File Reference . . . . .	1245

9.13.1 Detailed Description . . . . .	1247
9.13.2 Macro Definition Documentation . . . . .	1247
9.13.2.1 IMGDETAILS_LEN . . . . .	1247
9.13.2.2 MAX_BUILD_ID_LEN . . . . .	1247
9.13.2.3 MAX_CUST_ID_LEN . . . . .	1247
9.13.2.4 MAX_CUST_VALUE_LEN . . . . .	1247
9.13.2.5 MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH . . . . .	1247
9.13.2.6 MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH . . . . .	1247
9.13.2.7 MAX_FSN_LENGTH . . . . .	1247
9.13.2.8 UNIQUE_ID_LEN . . . . .	1247
9.13.3 Typedef Documentation . . . . .	1248
9.13.3.1 custFeaturesInfo . . . . .	1248
9.13.3.2 custFeaturesSetting . . . . .	1249
9.13.3.3 dmsCurrentPRLInfo . . . . .	1251
9.13.3.4 ERIFileparams . . . . .	1251
9.13.3.5 serialNumbersInfo . . . . .	1251
9.13.3.6 SLQSSwiGetHostDevInfoParams . . . . .	1252
9.13.3.7 SLQSSwiGetOSInfoParams . . . . .	1252
9.13.3.8 SLQSSwiGetSerialNoExtParams . . . . .	1253
9.13.3.9 SLQSSwiSetHostDevInfoParams . . . . .	1253
9.13.3.10 SLQSSwiSetOSInfoParams . . . . .	1254
9.13.4 Function Documentation . . . . .	1254
9.13.4.1 ActivateAutomatic . . . . .	1254
9.13.4.2 GetActivationState . . . . .	1255
9.13.4.3 GetDeviceCapabilities . . . . .	1255
9.13.4.4 GetFirmwareRevision . . . . .	1256
9.13.4.5 GetFirmwareRevisions . . . . .	1257
9.13.4.6 GetHardwareRevision . . . . .	1257
9.13.4.7 GetIMSI . . . . .	1258
9.13.4.8 GetManufacturer . . . . .	1258
9.13.4.9 GetModelID . . . . .	1259
9.13.4.10 GetNetworkTime . . . . .	1259
9.13.4.11 GetOfflineReason . . . . .	1260
9.13.4.12 GetPower . . . . .	1261
9.13.4.13 GetPRLVersion . . . . .	1261
9.13.4.14 GetSerialNumbers . . . . .	1262
9.13.4.15 GetVoiceNumber . . . . .	1262
9.13.4.16 ResetToFactoryDefaults . . . . .	1263
9.13.4.17 SetPower . . . . .	1263
9.13.4.18 SLQSDmsSwiGetResetInfo . . . . .	1264

9.13.4.19 SLQSDmsSwiIndicationRegister . . . . .	1264
9.13.4.20 SLQSGetBandCapabilities . . . . .	1264
9.13.4.21 SLQSGetBandCapability . . . . .	1265
9.13.4.22 SLQSGetCurrentPRLInfo . . . . .	1266
9.13.4.23 SLQSGetCustFeatures . . . . .	1266
9.13.4.24 SLQSGetCustFeaturesV2 . . . . .	1267
9.13.4.25 SLQSGetERIFile . . . . .	1267
9.13.4.26 SLQSGetSerialNumbers . . . . .	1268
9.13.4.27 SLQSSetCustFeatures . . . . .	1268
9.13.4.28 SLQSSetCustFeaturesV2 . . . . .	1268
9.13.4.29 SLQSSwiClearDyingGaspStatistics . . . . .	1269
9.13.4.30 SLQSSwiGetCrashAction . . . . .	1269
9.13.4.31 SLQSSwiGetCrashInfo . . . . .	1269
9.13.4.32 SLQSSwiGetDyingGaspCfg . . . . .	1270
9.13.4.33 SLQSSwiGetDyingGaspStatistics . . . . .	1270
9.13.4.34 SLQSSwiGetFirmwareCurr . . . . .	1270
9.13.4.35 SLQSSwiGetFSN . . . . .	1271
9.13.4.36 SLQSSwiGetFwUpdateStatus . . . . .	1271
9.13.4.37 SLQSSwiGetHostDevInfo . . . . .	1272
9.13.4.38 SLQSSwiGetOSInfo . . . . .	1272
9.13.4.39 SLQSSwiGetSerialNoExt . . . . .	1273
9.13.4.40 SLQSSwiGetUSBComp . . . . .	1273
9.13.4.41 SLQSSwiSetCrashAction . . . . .	1273
9.13.4.42 SLQSSwiSetDyingGaspCfg . . . . .	1274
9.13.4.43 SLQSSwiSetHostDevInfo . . . . .	1274
9.13.4.44 SLQSSwiSetOSInfo . . . . .	1275
9.13.4.45 SLQSSwiSetUSBComp . . . . .	1275
9.13.4.46 SLQSUIMGetState . . . . .	1276
9.13.4.47 UIMChangePIN . . . . .	1276
9.13.4.48 UIMGetControlKeyStatus . . . . .	1277
9.13.4.49 UIMGetICCID . . . . .	1278
9.13.4.50 UIMGetPINStatus . . . . .	1278
9.13.4.51 UIMSetControlKeyProtection . . . . .	1279
9.13.4.52 UIMSetPINProtection . . . . .	1280
9.13.4.53 UIMUnblockControlKey . . . . .	1281
9.13.4.54 UIMUnblockPIN . . . . .	1281
9.13.4.55 UIMVerifyPIN . . . . .	1282
9.13.4.56 ValidateSPC . . . . .	1283
9.14 qaGobiApiFms.h File Reference . . . . .	1283
9.14.1 Detailed Description . . . . .	1286

9.14.2	Macro Definition Documentation	1286
9.14.2.1	BUILD_ID_LEN	1286
9.14.2.2	DEVICE_OFFLINE	1286
9.14.2.3	DEVICE_RESET	1286
9.14.2.4	DEVICE_SHUTDOWN	1286
9.14.2.5	FIRMWARE_UPDATE_FAIL	1287
9.14.2.6	FIRMWARE_UPDATE_SUCCESS	1287
9.14.2.7	FIRMWARE_UPGRADE_SUCCESS	1287
9.14.2.8	G3K_FIRMWARE_DOWNLOAD	1287
9.14.2.9	GOBI_LISTENTRIES_MAX	1287
9.14.2.10	GOBI_MBN_BUILD_ID_STR_LEN	1287
9.14.2.11	GOBI_MBN_IMG_ID_STR_LEN	1287
9.14.2.12	GOBI_SET_IMG_PREF_RSPLN	1287
9.14.2.13	IMG_ID_LEN	1287
9.14.2.14	PRI_UPDATE_FAIL	1287
9.14.2.15	SLQSFWINFO_APPVERSION_SZ	1287
9.14.2.16	SLQSFWINFO_BOOTVERSION_SZ	1287
9.14.2.17	SLQSFWINFO_CARRIER_SZ	1287
9.14.2.18	SLQSFWINFO_CUR_CARR_NAME	1287
9.14.2.19	SLQSFWINFO_CUR_CARR_REV	1287
9.14.2.20	SLQSFWINFO_MODELID_SZ	1287
9.14.2.21	SLQSFWINFO_PACKAGEID_SZ	1287
9.14.2.22	SLQSFWINFO_PRIVERSION_SZ	1287
9.14.2.23	SLQSFWINFO_SKU_SZ	1287
9.14.2.24	SPKG_FIRMWARE_DOWNLOAD	1287
9.14.3	Enumeration Type Documentation	1287
9.14.3.1	eGobiDeviceSeries	1287
9.14.3.2	eGobiImageCarrier	1288
9.14.3.3	eGobiImageGPS	1289
9.14.3.4	eGobiImageRegion	1289
9.14.3.5	eGobiImageTech	1289
9.14.4	Function Documentation	1289
9.14.4.1	DeleteStoredImage	1289
9.14.4.2	eGetDeviceSeries	1290
9.14.4.3	GetImagesPreference	1290
9.14.4.4	GetImageStore	1291
9.14.4.5	GetStoredImages	1291
9.14.4.6	SetImagesPreference	1292
9.14.4.7	SLQSDownloadFirmwareToSlot	1292
9.14.4.8	SLQSGetBootVersionNumber	1293

9.14.4.9	SLQSGetFirmwareInfo	1294
9.14.4.10	SLQSGetImageInfo	1294
9.14.4.11	SLQSGetImageInfo_9x15	1295
9.14.4.12	SLQSGetImageInfoMC77xx	1296
9.14.4.13	SLQSGetImageInfoMC83xx	1296
9.14.4.14	SLQSGetValidFwPriCombinations	1297
9.14.4.15	SLQSIspkgFormatRequired	1297
9.14.4.16	SLQSSetSIMBasedImageSwitching	1297
9.14.4.17	SLQSSetSpkgFormatRequired	1298
9.14.4.18	SLQSSwiGetAllCarrierImages	1298
9.14.4.19	SLQSupgradeFirmware9x15	1299
9.14.4.20	upgrade_mc77xx_fw	1300
9.14.4.21	UpgradeFirmware2k	1300
9.15	qaGobiApilms.h File Reference	1300
9.15.1	Detailed Description	1301
9.15.2	Function Documentation	1301
9.15.2.1	SLQSGetIMSSMSConfig	1301
9.15.2.2	SLQSGetIMSUserConfig	1302
9.15.2.3	SLQSGetIMSVoIPConfig	1302
9.15.2.4	SLQSGetRegMgrConfig	1303
9.15.2.5	SLQSGetSIPConfig	1303
9.15.2.6	SLQSImsConfigIndicationRegister	1304
9.15.2.7	SLQSSetIMSSMSConfig	1304
9.15.2.8	SLQSSetIMSUserConfig	1305
9.15.2.9	SLQSSetIMSVoIPConfig	1305
9.15.2.10	SLQSSetRegMgrConfig	1306
9.15.2.11	SLQSSetSIPConfig	1306
9.16	qaGobiApiImsa.h File Reference	1307
9.16.1	Detailed Description	1307
9.16.2	Function Documentation	1307
9.16.2.1	SLQSGetIMSASRegStatus	1307
9.16.2.2	SLQSGetIMSASServiceStatus	1308
9.16.2.3	SLQSGetIMSASupportedFields	1308
9.16.2.4	SLQSGetIMSASupportedMsg	1309
9.16.2.5	SLQSRegisterIMSASIndication	1309
9.17	qaGobiApiLoc.h File Reference	1310
9.17.1	Detailed Description	1311
9.17.2	Macro Definition Documentation	1311
9.17.2.1	MAX_SENSOR_DATA_LEN	1311
9.17.2.2	MAX_TEMP_DATA_LEN	1311

9.17.3	Function Documentation	1311
9.17.3.1	SLQSLOCDelAssData	1311
9.17.3.2	SLQSLOCEventRegister	1311
9.17.3.3	SLQSLOCGetBestAvailPos	1312
9.17.3.4	SLQSLOCInjectPosition	1312
9.17.3.5	SLQSLOCInjectSensorData	1313
9.17.3.6	SLQSLOCInjectUTCtime	1313
9.17.3.7	SLQSLOCSetCradleMountConfig	1313
9.17.3.8	SLQSLOCSetExtPowerState	1314
9.17.3.9	SLQSLOCSetOpMode	1314
9.17.3.10	SLQSLOCStart	1315
9.17.3.11	SLQSLOCStop	1315
9.17.3.12	SwiLocGetAutoStart	1316
9.17.3.13	SwiLocSetAutoStart	1316
9.18	qaGobiApiNas.h File Reference	1316
9.18.1	Detailed Description	1321
9.18.2	Macro Definition Documentation	1321
9.18.2.1	IMSI_M_S1_LENGTH	1321
9.18.2.2	IMSI_M_S2_LENGTH	1321
9.18.2.3	MAX_DATA_SRV_CAPABILITIES	1321
9.18.2.4	MAX_DESCRIPTION_LENGTH	1321
9.18.2.5	MAX_PILOT_SETS	1321
9.18.2.6	MAX_SERV_SYSTEM_RADIO_INTERFACES	1321
9.18.2.7	NAM_NAME_LENGTH	1321
9.18.2.8	NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE	1322
9.18.2.9	NAS_SIG_INFO_MIN_dB_FLOAT_VALUE	1322
9.18.2.10	NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE	1322
9.18.2.11	PLMN_LENGTH	1322
9.18.2.12	SLQS_SS_INFO_LIST_MAX_ELEMENTS	1322
9.18.2.13	SLQS_SYSTEM_ID_SIZE	1322
9.18.2.14	UATISIZE	1322
9.18.3	Typedef Documentation	1322
9.18.3.1	SlqsNas3GppNetworkRAT	1322
9.18.3.2	slqsNetworkScanInfo	1322
9.18.3.3	sysSelectPrefInfo	1323
9.18.3.4	sysSelectPrefParams	1327
9.18.4	Enumeration Type Documentation	1333
9.18.4.1	_NAMS_RADIO_IF_TECHNOLOGY_	1333
9.18.4.2	eSYS_SRV_DOMAIN	1333
9.18.4.3	NAS_LTE_CPHY_CA_BW_NRB	1334

9.18.4.4	NAS_LTE_CPHY_SCELL_STATE	1334
9.18.5	Function Documentation	1334
9.18.5.1	GetACCOLC	1334
9.18.5.2	GetANAAAAAuthenticationStatus	1334
9.18.5.3	GetCDMANetworkParameters	1335
9.18.5.4	GetHomeNetwork	1337
9.18.5.5	GetHomeNetwork3GPP2	1337
9.18.5.6	GetNetworkPreference	1339
9.18.5.7	GetRFInfo	1340
9.18.5.8	GetServingNetwork	1340
9.18.5.9	GetServingNetworkCapabilities	1341
9.18.5.10	GetSignalStrengths	1342
9.18.5.11	InitiateDomainAttach	1343
9.18.5.12	InitiateNetworkRegistration	1343
9.18.5.13	PerformNetworkScan	1344
9.18.5.14	SetACCOLC	1344
9.18.5.15	SetCDMANetworkParameters	1345
9.18.5.16	SetNetworkPreference	1346
9.18.5.17	SLQSConfigSigInfo	1347
9.18.5.18	SLQSGetErrorRate	1348
9.18.5.19	SLQSGetNetworkTime	1348
9.18.5.20	SLQSGetOperatorNameData	1348
9.18.5.21	SLQSGetPLMNName	1349
9.18.5.22	SLQSGetServingSystem	1349
9.18.5.23	SLQSGetSignalStrength	1350
9.18.5.24	SLQSGetSysSelectionPref	1350
9.18.5.25	SLQSInitiateNetworkRegistration	1351
9.18.5.26	SLQSNasConfigSigInfo2	1351
9.18.5.27	SLQSNasGet3GPP2Subscription	1352
9.18.5.28	SLQSNasGetCellLocationInfo	1352
9.18.5.29	SLQSNasGetHDRColorCode	1353
9.18.5.30	SLQSNASGetLTECPHYCaInfo	1353
9.18.5.31	SLQSNasGetSigInfo	1353
9.18.5.32	SLQSNasGetSysInfo	1354
9.18.5.33	SLQSNasGetTxRxInfo	1354
9.18.5.34	SLQSNasIndicationRegister	1355
9.18.5.35	SLQSNasIndicationRegisterExt	1356
9.18.5.36	SLQSNasIndicationRegisterLTECphyCa	1356
9.18.5.37	SLQSNASSwiGetChannelLock	1356
9.18.5.38	SLQSNasSwiIndicationRegister	1357

9.18.5.39	SLQSNasSwiModemStatus	1357
9.18.5.40	SLQSNASSwiSetChannelLock	1358
9.18.5.41	SLQSPerformNetworkScan	1358
9.18.5.42	SLQSSetBandPreference	1359
9.18.5.43	SLQSSetSysSelectionPref	1360
9.18.5.44	SLQSSwiGetHDRPersonality	1360
9.18.5.45	SLQSSwiGetHDRProtSubtype	1361
9.18.5.46	SLQSSwiGetHRPDStats	1361
9.18.5.47	SLQSSwiGetLteCQI	1361
9.18.5.48	SLQSSwiNetworkDebug	1362
9.18.5.49	SLQSSwiPSDetach	1362
9.19	qaGobiApiOadm.h File Reference	1363
9.19.1	Detailed Description	1363
9.19.2	Function Documentation	1363
9.19.2.1	OMADMCancelSession	1363
9.19.2.2	OMADMGetPendingNIA	1363
9.19.2.3	OMADMGetSessionInfo	1364
9.19.2.4	OMADMStartSession	1365
9.20	qaGobiApiPds.h File Reference	1366
9.20.1	Detailed Description	1367
9.20.2	Macro Definition Documentation	1367
9.20.2.1	DEFAULTBYTEVALUE	1367
9.20.2.2	DEFAULTLONGVALUE	1367
9.20.2.3	DEFAULTWORDVALUE	1367
9.20.3	Enumeration Type Documentation	1367
9.20.3.1	anonymous enum	1367
9.20.4	Function Documentation	1367
9.20.4.1	ForceXTRADownload	1367
9.20.4.2	GetPDSDDefaults	1367
9.20.4.3	GetPDSSState	1368
9.20.4.4	GetPortAutomaticTracking	1369
9.20.4.5	GetServiceAutomaticTracking	1369
9.20.4.6	GetXTRAAutomaticDownload	1370
9.20.4.7	GetXTRANetwork	1370
9.20.4.8	GetXTRAValidity	1371
9.20.4.9	PDSInjectTimeReference	1371
9.20.4.10	ResetPDSDData	1372
9.20.4.11	SetPDSDDefaults	1373
9.20.4.12	SetPDSSState	1373
9.20.4.13	SetPortAutomaticTracking	1374

9.20.4.14	SetServiceAutomaticTracking	1374
9.20.4.15	SetXTRAAutomaticDownload	1375
9.20.4.16	SetXTRANetwork	1375
9.20.4.17	SLQSGetAGPSConfig	1376
9.20.4.18	SLQSGetGPSStateInfo	1376
9.20.4.19	SLQSPDSDeterminePosition	1377
9.20.4.20	SLQSPDSInjectAbsoluteTimeReference	1377
9.20.4.21	SLQSPDSInjectPositionData	1378
9.20.4.22	SLQSSetAGPSConfig	1378
9.20.4.23	SLQSSetPositionMethodState	1379
9.20.4.24	StartPDSTrackingSessionExt	1379
9.20.4.25	StopPDSTrackingSession	1380
9.21	qaGobiApiQos.h File Reference	1381
9.21.1	Detailed Description	1381
9.21.2	Macro Definition Documentation	1382
9.21.2.1	MAX_QOS_FILTER_TLV	1382
9.21.2.2	MAX_QOS_SPEC_PER_APN	1382
9.21.3	Function Documentation	1382
9.21.3.1	SLQSQosGetFlowStatus	1382
9.21.3.2	SLQSQosGetGranted	1382
9.21.3.3	SLQSQosGetNetworkStatus	1383
9.21.3.4	SLQSQosGetNWProf	1383
9.21.3.5	SLQSQosModify	1383
9.21.3.6	SLQSQosRel	1384
9.21.3.7	SLQSQosReq	1384
9.21.3.8	SLQSQosReset	1385
9.21.3.9	SLQSQosResume	1385
9.21.3.10	SLQSQosSuspend	1386
9.21.3.11	SLQSQosSwiReadApnExtraParams	1386
9.21.3.12	SLQSQosSwiReadDataStats	1387
9.22	qaGobiApiRms.h File Reference	1387
9.22.1	Detailed Description	1387
9.22.2	Function Documentation	1387
9.22.2.1	GetSMSWake	1387
9.22.2.2	SetSMSWake	1388
9.23	qaGobiApiSar.h File Reference	1388
9.23.1	Detailed Description	1389
9.23.2	Enumeration Type Documentation	1389
9.23.2.1	eQMISARRFState	1389
9.23.3	Function Documentation	1390

9.23.3.1	SLQSGetRfSarState	1390
9.23.3.2	SLQSSetRfSarState	1390
9.24	qaGobiApiSms.h File Reference	1391
9.24.1	Detailed Description	1393
9.24.2	Macro Definition Documentation	1393
9.24.2.1	ABSOLUTE_VALIDITY	1393
9.24.2.2	CONFIG_LEN	1393
9.24.2.3	MAX_SMS_ROUTES	1393
9.24.2.4	NUM_OF_SET	1393
9.24.2.5	TIME_DATE_BUF	1393
9.24.2.6	TIME_STAMP_BUF	1393
9.24.3	Typedef Documentation	1393
9.24.3.1	getIndicationRegResp	1393
9.24.3.2	getTransLayerInfoResp	1394
9.24.3.3	getTransNWRegInfoResp	1394
9.24.3.4	qaQmi3GPP2BroadcastCfgInfo	1395
9.24.3.5	qaQmi3GPPBroadcastCfgInfo	1395
9.24.3.6	setIndicationRegReq	1395
9.24.3.7	transLayerInfo	1396
9.24.4	Function Documentation	1396
9.24.4.1	GetSMSCAddress	1396
9.24.4.2	SaveSMS	1397
9.24.4.3	SendSMS	1398
9.24.4.4	SetSMSCAddress	1399
9.24.4.5	SLQSCDMADecodeMTTextMsg	1399
9.24.4.6	SLQSCDMAEncodeMOTextMsg	1399
9.24.4.7	SLQSDeleteSMS	1400
9.24.4.8	SLQSGetIndicationRegister	1401
9.24.4.9	SLQSGetMessageWaiting	1401
9.24.4.10	SLQSGetSMS	1402
9.24.4.11	SLQSGetSmsBroadcastConfig	1403
9.24.4.12	SLQSGetSMSList	1403
9.24.4.13	SLQSGetTransLayerInfo	1404
9.24.4.14	SLQSGetTransNWRegInfo	1405
9.24.4.15	SLQSModifySMSStatus	1405
9.24.4.16	SLQSSendAsyncSMS	1406
9.24.4.17	SLQSSendLongSMS	1406
9.24.4.18	SLQSSendSMS	1407
9.24.4.19	SLQSSetIndicationRegister	1408
9.24.4.20	SLQSSetSmsBroadcastActivation	1408

9.24.4.21	SLQSSetSmsBroadcastConfig	1409
9.24.4.22	SLQSSetSmsStorage	1409
9.24.4.23	SLQSSmsGetMaxStorageSize	1410
9.24.4.24	SLQSSmsGetMessageProtocol	1410
9.24.4.25	SLQSSmsSetRoutes	1411
9.24.4.26	SLQSSwiGetSMSStorage	1411
9.24.4.27	SLQSWCDMADecodeLongTextMsg	1412
9.24.4.28	SLQSWCDMADecodeMTTextMsg	1412
9.24.4.29	SLQSWCDMAEncodeMOTextMsg	1413
9.25	qaGobiApiSwi.h File Reference	1413
9.25.1	Detailed Description	1413
9.25.2	Function Documentation	1413
9.25.2.1	SLQSGetPidof	1413
9.25.2.2	SLQSGetSdkVersion	1414
9.25.2.3	SLQSSendRawQMI	1414
9.26	qaGobiApiSwiAudio.h File Reference	1414
9.26.1	Detailed Description	1415
9.26.2	Macro Definition Documentation	1415
9.26.2.1	MAX_LEN_IFACE_TABLE	1415
9.26.3	Function Documentation	1415
9.26.3.1	SLQSGetM2MAudioProfile	1415
9.26.3.2	SLQSGetM2MAudioVolume	1416
9.26.3.3	SLQSGetM2MAVMute	1416
9.26.3.4	SLQSGetM2MSpkrGain	1417
9.26.3.5	SLQSSetM2MAudioAVCFG	1417
9.26.3.6	SLQSSetM2MAudioLPBK	1417
9.26.3.7	SLQSSetM2MAudioNVDef	1418
9.26.3.8	SLQSSetM2MAudioProfile	1418
9.26.3.9	SLQSSetM2MAudioVolume	1418
9.26.3.10	SLQSSetM2MAVMute	1419
9.26.3.11	SLQSSetM2MSpkrGain	1419
9.27	qaGobiApiSwiOmadms.h File Reference	1420
9.27.1	Detailed Description	1420
9.27.2	Typedef Documentation	1421
9.27.2.1	SLQSOMADMSessionInfo	1421
9.27.2.2	SLQSOMADMSettings	1422
9.27.2.3	SLQSOMADMSettingsReqParams	1423
9.27.2.4	SLQSOMADMSettingsReqParams3	1424
9.27.3	Function Documentation	1425
9.27.3.1	SLQSOMADMCancelSession	1425

9.27.3.2	SLQSOMADMGetSessionInfo	1425
9.27.3.3	SLQSOMADMGetSettings	1426
9.27.3.4	SLQSOMADMGetSettings2	1426
9.27.3.5	SLQSOMADMSendSelection	1427
9.27.3.6	SLQSOMADMSendSelection2	1427
9.27.3.7	SLQSOMADMSetSettings	1428
9.27.3.8	SLQSOMADMSetSettings2	1429
9.27.3.9	SLQSOMADMSetSettings3	1429
9.27.3.10	SLQSOMADMStartSession	1429
9.27.3.11	SLQSOMADMStartSession2	1430
9.28	qaGobiApiTableBandClasses.h File Reference	1430
9.28.1	Detailed Description	1430
9.28.2	Band Classes (Value - Description)	1430
9.28.2.1	LTE Bands	1432
9.29	qaGobiApiTableCallControlReturnReasons.h File Reference	1433
9.29.1	Detailed Description	1433
9.29.2	Call Control Result Reasons (Value - Name - Description)	1433
9.30	qaGobiApiTableCallEndReasons.h File Reference	1434
9.30.1	Detailed Description	1434
9.30.2	Call end reason codes (Code - Reason)	1434
9.30.2.1	Technology-agnostic call end reasons	1434
9.30.2.2	EVDO CDMA 1xEV-DO	1435
9.30.2.3	WCDMA/GSM call end reasons	1435
9.30.2.4	EVDO CDMA 1xEV-DO	1437
9.30.2.5	call end reason type	1438
9.30.2.6	Mobile IP call end reasons (Type=1)	1438
9.30.2.7	Internal call end reasons (Type=2)	1440
9.30.2.8	Call Manager defined call end reasons (Type=3)	1441
9.30.2.9	3GPP specification defined call end reasons (Type=6)	1446
9.30.2.10	PPP call end reasons (Type=7)	1448
9.30.2.11	EHRPD call end reasons (Type=8)	1448
9.30.2.12	IPv6 call end reasons (Type=9)	1449
9.31	qaGobiApiTableCarrierCodes.h File Reference	1449
9.31.1	Detailed Description	1449
9.31.2	Carrier Codes (Number - Carrier)	1449
9.32	qaGobiApiTableCodingScheme.h File Reference	1451
9.32.1	Detailed Description	1451
9.32.2	Call Control Result Reasons (Value - Name - Description)	1451
9.32.2.1	Use of bits 3..0	1451
9.32.3	Coding Group Bits 7..4(0001)	1452

9.32.3.1	use of bits 3..0	1452
9.32.4	Coding Group Bits 7..4(0010)	1452
9.32.4.1	use of bits 3..0	1452
9.32.5	Coding Group Bits 7..4(0011)	1452
9.32.5.1	use of bits 3..0	1452
9.32.6	Coding Group Bits 7..4(01xx)	1453
9.32.6.1	use of bits 3..0	1453
9.32.7	Coding Group Bits 7..4(1001)	1453
9.32.7.1	Reserved coding groups	1453
9.32.8	Coding Group Bits 7..4(1010..1101)	1453
9.32.8.1	Reserved coding groups	1453
9.32.9	Coding Group Bits 7..4(1110)	1453
9.32.9.1	Defined by the WAP Forum	1453
9.32.10	Coding Group Bits 7..4 (1111)	1453
9.32.10.1	Data coding / message handling	1454
9.32.11	Macro Definition Documentation	1454
9.32.11.1	__GOBI_API_CODING_SCHEME_H__	1454
9.33	qaGobiApiTableGpsCapabilityCodes.h File Reference	1454
9.33.1	Detailed Description	1454
9.33.2	GPS capability (Value - Capability)	1454
9.34	qaGobiApiTablePowerModes.h File Reference	1454
9.34.1	Detailed Description	1454
9.34.2	Power Modes (Value - Description)	1455
9.35	qaGobiApiTableRadioInterfaces.h File Reference	1455
9.35.1	Detailed Description	1455
9.35.2	Radio interface	1455
9.35.2.1	Technology (Value - Radio Interface Technology)	1455
9.36	qaGobiApiTableRegionCodes.h File Reference	1456
9.36.1	Detailed Description	1456
9.36.2	Region Codes (Code - Region)	1456
9.37	qaGobiApiTableServiceOptions.h File Reference	1456
9.37.1	Detailed Description	1456
9.37.2	Service Option codes (Code - Reason)	1456
9.37.2.1	Description	1456
9.38	qaGobiApiTableSupServiceInfoClasses.h File Reference	1458
9.38.1	Detailed Description	1458
9.38.2	Supplementary Service Information Classes (Value - Service Class)	1459
9.39	qaGobiApiTableSwiAudio.h File Reference	1459
9.39.1	Detailed Description	1459
9.39.2	ACDB Device (Device ID - description)	1459

9.39.3	Physical Interface (Device ID - description - Interface parameters)	1459
9.40	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	1459
9.40.1	Detailed Description	1460
9.40.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	1460
9.41	qaGobiApiTableVoiceCallEndReasons.h File Reference	1461
9.41.1	Detailed Description	1461
9.41.2	Voice Call and supplementary services end reason codes (Code - Reason)	1461
9.41.2.1	General	1461
9.41.2.2	service Errors	1463
9.41.2.3	control cause values	1464
9.41.2.4	reject causes	1465
9.41.2.5	reject causes	1466
9.41.2.6	reject causes	1466
9.41.2.7	stratum reject causes	1466
9.41.2.8	reject causes	1467
9.41.2.9	IP end reasons	1467
9.42	qaGobiApiTmd.h File Reference	1467
9.42.1	Detailed Description	1468
9.42.2	Macro Definition Documentation	1468
9.42.2.1	MAX_MITIGATION_DEV_ID_LEN	1468
9.42.2.2	MAX_MITIGATION_DEV_LIST_LEN	1468
9.42.3	Function Documentation	1468
9.42.3.1	SLQSTmdDeRegNotMitigationLvl	1468
9.42.3.2	SLQSTmdGetMitigationDevList	1468
9.42.3.3	SLQSTmdGetMitigationLvl	1469
9.42.3.4	SLQSTmdRegNotMitigationLvl	1469
9.43	qaGobiApiUim.h File Reference	1469
9.43.1	Detailed Description	1471
9.43.2	Macro Definition Documentation	1472
9.43.2.1	MAX_ACTIVE_PERS_FEATURES	1472
9.43.2.2	MAX_CONTENT_LENGTH	1472
9.43.2.3	MAX_DESCRIPTION_LENGTH	1472
9.43.2.4	MAX_ICCID_LENGTH	1472
9.43.2.5	MAX_NO_OF_APPLICATIONS	1472
9.43.2.6	MAX_NO_OF_SLOTS	1472
9.43.2.7	MAX_PATH_LENGTH	1472
9.43.2.8	MAX_PUK_LENGTH	1472
9.43.2.9	MAX_SLOTS_STATUS	1472
9.43.3	Function Documentation	1472
9.43.3.1	SLQSUIAuthenticate	1472

9.43.3.2	SLQSUIMChangePin	1472
9.43.3.3	SLQSUIMDepersonalization	1473
9.43.3.4	SLQSUIMEventRegister	1474
9.43.3.5	SLQSUIMGetCardStatus	1474
9.43.3.6	SLQSUIMGetConfiguration	1475
9.43.3.7	SLQSUIMGetFileAttributes	1475
9.43.3.8	SLQSUIMGetSlotsStatus	1476
9.43.3.9	SLQSUIMPowerDown	1476
9.43.3.10	SLQSUIMPowerUp	1476
9.43.3.11	SLQSUIMReadTransparent	1477
9.43.3.12	SLQSUIMRefreshComplete	1477
9.43.3.13	SLQSUIMRefreshGetLastEvent	1478
9.43.3.14	SLQSUIMRefreshOK	1479
9.43.3.15	SLQSUIMRefreshRegister	1479
9.43.3.16	SLQSUIMReset	1480
9.43.3.17	SLQSUIMSetPinProtection	1480
9.43.3.18	SLQSUIMSwitchSlot	1481
9.43.3.19	SLQSUIMUnblockPin	1481
9.43.3.20	SLQSUIMVerifyPin	1482
9.44	qaGobiApiVoice.h File Reference	1482
9.44.1	Detailed Description	1485
9.44.2	Macro Definition Documentation	1485
9.44.2.1	MAX_CALL_NO_LEN	1485
9.44.2.2	MAX_DESCRIPTION_LENGTH	1485
9.44.2.3	MAX_NO_OF_CALLS	1485
9.44.2.4	MAXUSSDLENGTH	1485
9.44.2.5	PASSWORD_LENGTH	1485
9.44.3	Enumeration Type Documentation	1485
9.44.3.1	serviceClassInformation	1485
9.44.4	Function Documentation	1486
9.44.4.1	AnswerUSSD	1486
9.44.4.2	CancelUSSD	1486
9.44.4.3	OriginateUSSD	1487
9.44.4.4	SLQSOriginateUSSD	1487
9.44.4.5	SLQSVoiceALSSelectLine	1487
9.44.4.6	SLQSVoiceALSSetLineSwitching	1488
9.44.4.7	SLQSVoiceAnswerCall	1488
9.44.4.8	SLQSVoiceBindSubscription	1489
9.44.4.9	SLQSVoiceBurstDTMF	1489
9.44.4.10	SLQSVoiceDialCall	1490

9.44.4.11 SLQSVoiceEndCall . . . . .	1490
9.44.4.12 SLQSVoiceGetAllCallInfo . . . . .	1491
9.44.4.13 SLQSVoiceGetCallBarring . . . . .	1491
9.44.4.14 SLQSVoiceGetCallForwardingStatus . . . . .	1492
9.44.4.15 SLQSVoiceGetCallInfo . . . . .	1492
9.44.4.16 SLQSVoiceGetCallWaiting . . . . .	1493
9.44.4.17 SLQSVoiceGetCLIP . . . . .	1493
9.44.4.18 SLQSVoiceGetCLIR . . . . .	1494
9.44.4.19 SLQSVoiceGetCNAP . . . . .	1495
9.44.4.20 SLQSVoiceGetCOLP . . . . .	1495
9.44.4.21 SLQSVoiceGetCOLR . . . . .	1496
9.44.4.22 SLQSVoiceGetConfig . . . . .	1496
9.44.4.23 SLQSVoiceIndicationRegister . . . . .	1497
9.44.4.24 SLQSVoiceManageCalls . . . . .	1497
9.44.4.25 SLQSVoiceOrigUSSDNoWait . . . . .	1498
9.44.4.26 SLQSVoiceSendFlash . . . . .	1498
9.44.4.27 SLQSVoiceSetCallBarringPassword . . . . .	1499
9.44.4.28 SLQSVoiceSetConfig . . . . .	1499
9.44.4.29 SLQSVoiceSetPreferredPrivacy . . . . .	1500
9.44.4.30 SLQSVoiceSetSUPSService . . . . .	1500
9.44.4.31 SLQSVoiceStartContDTMF . . . . .	1501
9.44.4.32 SLQSVoiceStopContDTMF . . . . .	1501
9.45 qaGobiApiWds.h File Reference . . . . .	1502
9.45.1 Detailed Description . . . . .	1505
9.45.2 Macro Definition Documentation . . . . .	1506
9.45.2.1 IPV6_ADDRESS_ARRAY_SIZE . . . . .	1506
9.45.3 Typedef Documentation . . . . .	1506
9.45.3.1 GetProfileSettingIn . . . . .	1506
9.45.3.2 GetProfileSettingOut . . . . .	1506
9.45.3.3 QmiProfileInfo . . . . .	1506
9.45.3.4 QmiWDSDataBearers . . . . .	1506
9.45.3.5 QmiWDSDataBearerTechnology . . . . .	1507
9.45.3.6 slqs3GPPConfigItem . . . . .	1508
9.45.4 Enumeration Type Documentation . . . . .	1509
9.45.4.1 qmiDataBearerMasks . . . . .	1509
9.45.5 Function Documentation . . . . .	1509
9.45.5.1 GetAutoconnect . . . . .	1509
9.45.5.2 GetByteTotals . . . . .	1509
9.45.5.3 GetConnectionRate . . . . .	1510
9.45.5.4 GetDataBearerTechnology . . . . .	1511

9.45.5.5	GetDefaultProfile	1511
9.45.5.6	GetDefaultProfileLTE	1513
9.45.5.7	GetDefaultProfileNum	1514
9.45.5.8	GetDormancyState	1515
9.45.5.9	GetIPAddressLTE	1515
9.45.5.10	GetLastMobileIPError	1516
9.45.5.11	GetMobileIP	1516
9.45.5.12	GetMobileIPProfile	1517
9.45.5.13	GetPacketStatistics	1518
9.45.5.14	GetPacketStatus	1519
9.45.5.15	GetSessionDuration	1520
9.45.5.16	GetSessionState	1520
9.45.5.17	iGetByteTotals	1521
9.45.5.18	iGetConnectionRate	1521
9.45.5.19	iGetPacketStatistics	1521
9.45.5.20	iSLQSMISetIPFamilyPreference	1521
9.45.5.21	RMSetTransferStatistics	1521
9.45.5.22	SetActiveMobileIPProfile	1521
9.45.5.23	SetAutoconnect	1522
9.45.5.24	SetDefaultProfile	1522
9.45.5.25	SetDefaultProfileLTE	1523
9.45.5.26	SetDefaultProfileLTEV2	1525
9.45.5.27	SetDefaultProfileNum	1526
9.45.5.28	SetMobileIP	1527
9.45.5.29	SetMobileIPParameters	1527
9.45.5.30	SetMobileIPProfile	1528
9.45.5.31	SLQSAutoConnect	1529
9.45.5.32	SLQSCreateProfile	1530
9.45.5.33	SLQSDeleteProfile	1530
9.45.5.34	SLQSGet3GPPConfigItem	1531
9.45.5.35	SLQSGetByteTotals	1531
9.45.5.36	SLQSGetConnectionRate	1532
9.45.5.37	SLQSGetCurrDataSystemStat	1532
9.45.5.38	SLQSGetCurrentChannelRate	1533
9.45.5.39	SLQSGetDataBearerTechnology	1533
9.45.5.40	SLQSGetDataBearerTechnologyExt	1534
9.45.5.41	SLQSGetDUNCallInfo	1534
9.45.5.42	SLQSGetPacketStatistics	1535
9.45.5.43	SLQSGetProfile	1535
9.45.5.44	SLQSGetProfileSettings	1537

9.45.5.45 SLQSGetRuntimeSettings . . . . .	1537
9.45.5.46 SLQSGetSessionState . . . . .	1537
9.45.5.47 SLQSModifyProfile . . . . .	1538
9.45.5.48 SLQSResetPacketStatics . . . . .	1539
9.45.5.49 SLQSSet3GPPConfigItem . . . . .	1539
9.45.5.50 SLQSSetProfile . . . . .	1539
9.45.5.51 SLQSSetDHCPv4ClientConfig . . . . .	1541
9.45.5.52 SLQSSetLoopback . . . . .	1541
9.45.5.53 SLQSSetDHCPv4ClientConfig . . . . .	1541
9.45.5.54 SLQSSetLoopback . . . . .	1542
9.45.5.55 SLQSStartStopDataSession . . . . .	1542
9.45.5.56 SLQSWdsGoActive . . . . .	1543
9.45.5.57 SLQSWdsGoDormant . . . . .	1543
9.45.5.58 SLQSWdsSetEventReport . . . . .	1543
9.45.5.59 SLQSWdsSwiPDPRuntimeSettings . . . . .	1544
9.45.5.60 WDS_IsGobiDevice . . . . .	1544
9.46 qaNasGetRFBandInfo.h File Reference . . . . .	1544
9.46.1 Enumeration Type Documentation . . . . .	1545
9.46.1.1 eQMI_NAS_GET_RF_INFO_RESP . . . . .	1545
9.46.2 Function Documentation . . . . .	1545
9.46.2.1 PkQmiNasGetRFBandInfo . . . . .	1545
9.46.2.2 UpkQmiNasGetRFBandInfo . . . . .	1545
9.47 qaNasPerformNetworkScan.h File Reference . . . . .	1545
9.47.1 Macro Definition Documentation . . . . .	1546
9.47.1.1 FORBIDDEN_INDEX . . . . .	1546
9.47.1.2 INDEX_ZERO . . . . .	1546
9.47.1.3 MAX_DESCRIPTION_LENGTH . . . . .	1546
9.47.1.4 PREFERRED_INDEX . . . . .	1546
9.47.1.5 QMI_NAS_MAX_INSTANCES . . . . .	1546
9.47.1.6 QMI_NAS_NETSTATUS_MASK . . . . .	1546
9.47.1.7 ROAMING_INDEX . . . . .	1546
9.47.2 Enumeration Type Documentation . . . . .	1546
9.47.2.1 eQMI_NAS_PERFORM_NETWORK_SCAN_RESP . . . . .	1546
9.47.3 Function Documentation . . . . .	1546
9.47.3.1 PkQmiNasPerformNetworkScan . . . . .	1546
9.47.3.2 UpkQmiNasPerformNetworkScan . . . . .	1546
9.48 qmerrno.h File Reference . . . . .	1546
9.48.1 Enumeration Type Documentation . . . . .	1548
9.48.1.1 eQCWWANError . . . . .	1548
9.48.1.2 qm_wds_ds_profile_extended_err_codes . . . . .	1553

9.49 qos.h File Reference	1553
9.49.1 Macro Definition Documentation	1555
9.49.1.1 LIBPACK_MAX_QOS_FILTERS	1555
9.49.1.2 LIBPACK_MAX_QOS_FLOW_PER_APN_STATS	1555
9.49.1.3 LIBPACK_MAX_QOS_FLOWS	1555
9.49.2 Function Documentation	1555
9.49.2.1 pack_qos_SLQSQosGetNetworkStatus	1555
9.49.2.2 pack_qos_SLQSQosSwiReadApnExtraParams	1555
9.49.2.3 pack_qos_SLQSQosSwiReadDataStats	1556
9.49.2.4 pack_qos_SLQSSetQosEventCallback	1557
9.49.2.5 unpack_qos_SLQSQosGetNetworkStatus	1557
9.49.2.6 unpack_qos_SLQSQosSwiReadApnExtraParams	1558
9.49.2.7 unpack_qos_SLQSQosSwiReadDataStats	1558
9.49.2.8 unpack_qos_SLQSSetQosEventCallback	1559
9.49.2.9 unpack_qos_SLQSSetQosEventCallback_ind	1559
9.49.2.10 unpack_qos_SLQSSetQosNWStatusCallback_ind	1560
9.49.2.11 unpack_qos_SLQSSetQosPriEventCallback_ind	1560
9.49.2.12 unpack_qos_SLQSSetQosStatusCallback_ind	1561
9.50 sms.h File Reference	1561
9.50.1 Macro Definition Documentation	1563
9.50.1.1 MAX_CDMA_ENC_MO_TXT_MSG_SIZE	1563
9.50.1.2 MAX_MS_TRANSFER_ROUTE_MSG	1563
9.50.1.3 MAX_MSC_ADDRESS_SIZE	1563
9.50.1.4 MAX_MSE_TWS_MSG	1563
9.50.1.5 MAX_SMS_LIST_SIZE	1563
9.50.1.6 MAX_SMS_MESSAGE_SIZE	1563
9.50.2 Typedef Documentation	1563
9.50.2.1 sMSCAddressInfo	1563
9.50.2.2 sMSEtwsMessageInfo	1564
9.50.2.3 sMSEtwsPlmnInfo	1564
9.50.2.4 sMSMessageModelInfo	1564
9.50.2.5 sMSMTMessageInfo	1564
9.50.2.6 sMSOnIMSInfo	1564
9.50.2.7 sMSTransferRouteMTMessageInfo	1564
9.50.3 Enumeration Type Documentation	1565
9.50.3.1 eqmiCbkJSetStatus	1565
9.50.4 Function Documentation	1565
9.50.4.1 pack_sms_SendSMS	1565
9.50.4.2 pack_sms_SetNewSMSCallback	1565
9.50.4.3 pack_sms_SLQSDDeleteSMS	1566

9.50.4.4	<a href="#">pack_sms_SLQSGetSMS</a>	1566
9.50.4.5	<a href="#">pack_sms_SLQSGetSMSList</a>	1567
9.50.4.6	<a href="#">pack_sms_SLQSModifySMSStatus</a>	1567
9.50.4.7	<a href="#">unpack_sms_SendSMS</a>	1567
9.50.4.8	<a href="#">unpack_sms_SetNewSMSCallback</a>	1568
9.50.4.9	<a href="#">unpack_sms_SetNewSMSCallback_ind</a>	1568
9.50.4.10	<a href="#">unpack_sms_SLQSDeleteSMS</a>	1568
9.50.4.11	<a href="#">unpack_sms_SLQSGetSMS</a>	1569
9.50.4.12	<a href="#">unpack_sms_SLQSGetSMSList</a>	1569
9.50.4.13	<a href="#">unpack_sms_SLQSModifySMSStatus</a>	1569
9.50.4.14	<a href="#">unpack_sms_SLQSWmsMemoryFullCallBack_ind</a>	1570
9.51	<a href="#">SwiDataTypes.h File Reference</a>	1570
9.51.1	<a href="#">Detailed Description</a>	1571
9.51.2	<a href="#">Macro Definition Documentation</a>	1571
9.51.2.1	<a href="#">QMI_NO_LTE_FW_SUPPORT</a>	1571
9.51.2.2	<a href="#">QMI_TLV_PLACEHOLDER</a>	1571
9.51.2.3	<a href="#">SWI_API</a>	1571
9.51.2.4	<a href="#">UNUSEDPARAM</a>	1571
9.51.3	<a href="#">Typedef Documentation</a>	1571
9.51.3.1	<a href="#">BOOL</a>	1571
9.51.3.2	<a href="#">BYTE</a>	1571
9.51.3.3	<a href="#">CHAR</a>	1571
9.51.3.4	<a href="#">FLOAT</a>	1571
9.51.3.5	<a href="#">INT32</a>	1571
9.51.3.6	<a href="#">INT8</a>	1571
9.51.3.7	<a href="#">LPCSTR</a>	1571
9.51.3.8	<a href="#">SHORT</a>	1571
9.51.3.9	<a href="#">ULONG</a>	1571
9.51.3.10	<a href="#">ULONGLONG</a>	1571
9.51.3.11	<a href="#">USHORT</a>	1571
9.51.3.12	<a href="#">WORD</a>	1571
9.52	<a href="#">swiloc.h File Reference</a>	1571
9.52.1	<a href="#">Function Documentation</a>	1572
9.52.1.1	<a href="#">pack_swiloc_SwiLocGetAutoStart</a>	1572
9.52.1.2	<a href="#">pack_swiloc_SwiLocSetAutoStart</a>	1572
9.52.1.3	<a href="#">unpack_swiloc_SwiLocGetAutoStart</a>	1572
9.52.1.4	<a href="#">unpack_swiloc_SwiLocSetAutoStart</a>	1573
9.53	<a href="#">swioma.h File Reference</a>	1573
9.53.1	<a href="#">Macro Definition Documentation</a>	1574
9.53.1.1	<a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a>	1574

9.53.2	Function Documentation	1574
9.53.2.1	pack_swioma_SLQSOMADMAAlertCallback	1574
9.53.2.2	pack_swioma_SLQSOMADMCancelSession	1575
9.53.2.3	pack_swioma_SLQSOMADMGetSessionInfo	1575
9.53.2.4	pack_swioma_SLQSOMADMGetSettings	1576
9.53.2.5	pack_swioma_SLQSOMADMSendSelection	1577
9.53.2.6	pack_swioma_SLQSOMADMSetSettings	1577
9.53.2.7	pack_swioma_SLQSOMADMStartSession	1578
9.53.2.8	unpack_swioma_SLQSOMADMAAlertCallback	1578
9.53.2.9	unpack_swioma_SLQSOMADMAAlertCallback_ind	1579
9.53.2.10	unpack_swioma_SLQSOMADMCancelSession	1579
9.53.2.11	unpack_swioma_SLQSOMADMGetSessionInfo	1580
9.53.2.12	unpack_swioma_SLQSOMADMGetSettings	1580
9.53.2.13	unpack_swioma_SLQSOMADMSendSelection	1581
9.53.2.14	unpack_swioma_SLQSOMADMSetSettings	1581
9.53.2.15	unpack_swioma_SLQSOMADMStartSession	1582
9.54	SWIWWANCMAPI.h File Reference	1582
9.55	uim.h File Reference	1582
9.55.1	Macro Definition Documentation	1584
9.55.1.1	MAX_DESCRIPTION_LENGTH	1584
9.55.1.2	MAX_ICCID_LENGTH	1584
9.55.1.3	MAX_NO_OF_APPLICATIONS	1584
9.55.1.4	MAX_NO_OF_SLOTS	1584
9.55.1.5	MAX_SLOTS_STATUS	1584
9.55.1.6	UIM_MAX_DESCRIPTION_LENGTH	1584
9.55.1.7	UIM_MAX_NO_OF_APPLICATIONS	1584
9.55.1.8	UIM_MAX_NO_OF_SLOTS	1584
9.55.1.9	UIM_UINT8_MAX_STRING_SZ	1584
9.55.2	Function Documentation	1584
9.55.2.1	pack_uim_ChangePin	1584
9.55.2.2	pack_uim_GetCardStatus	1584
9.55.2.3	pack_uim_ReadTransparent	1585
9.55.2.4	pack_uim_SetPinProtection	1585
9.55.2.5	pack_uim_SLQSUIEventRegister	1586
9.55.2.6	pack_uim_SLQSUIGetSlotsStatus	1586
9.55.2.7	pack_uim_SLQSUISSwitchSlot	1586
9.55.2.8	pack_uim_UnblockPin	1587
9.55.2.9	pack_uim_VerifyPin	1587
9.55.2.10	unpack_uim_ChangePin	1587
9.55.2.11	unpack_uim_GetCardStatus	1588

9.55.2.12	unpack_uim_ReadTransparent	1588
9.55.2.13	unpack_uim_SetPinProtection	1588
9.55.2.14	unpack_uim_SetUimSlotStatusChangeCallback_ind	1589
9.55.2.15	unpack_uim_SLQSUIEventRegister	1589
9.55.2.16	unpack_uim_SLQSUIGetSlotsStatus	1589
9.55.2.17	unpack_uim_SLQSUISetStatusChangeCallBack_ind	1590
9.55.2.18	unpack_uim_SLQSUIMSwitchSlot	1590
9.55.2.19	unpack_uim_UnblockPin	1591
9.55.2.20	unpack_uim_VerifyPin	1591
9.56	wds.h File Reference	1591
9.56.1	Macro Definition Documentation	1595
9.56.1.1	IPV6_ADDRESS_ARRAY_SIZE	1595
9.56.1.2	MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	1595
9.56.1.3	PACK_WDS_IPV4	1595
9.56.1.4	PACK_WDS_IPV6	1595
9.56.2	Typedef Documentation	1595
9.56.2.1	UnpackQmiProfileInfo	1595
9.56.3	Function Documentation	1595
9.56.3.1	pack_wds_GetConnectionRate	1596
9.56.3.2	pack_wds_GetDefaultProfile	1596
9.56.3.3	pack_wds_GetDefaultProfileNum	1596
9.56.3.4	pack_wds_GetDormancyState	1597
9.56.3.5	pack_wds_GetLastMobileIPError	1597
9.56.3.6	pack_wds_GetMobileIP	1598
9.56.3.7	pack_wds_GetMobileIPProfile	1598
9.56.3.8	pack_wds_GetPacketStatus	1598
9.56.3.9	pack_wds_GetSessionDuration	1599
9.56.3.10	pack_wds_GetSessionState	1599
9.56.3.11	pack_wds_RMSetTransferStatistics	1600
9.56.3.12	pack_wds_SetDefaultProfile	1600
9.56.3.13	pack_wds_SetDefaultProfileNum	1600
9.56.3.14	pack_wds_SetMobileIPProfile	1601
9.56.3.15	pack_wds_SLQSCreateProfile	1601
9.56.3.16	pack_wds_SLQSDeleteProfile	1602
9.56.3.17	pack_wds_SLQSGet3GPPConfigItem	1602
9.56.3.18	pack_wds_SLQSGetCurrDataSystemStat	1602
9.56.3.19	pack_wds_SLQSGetDataBearerTechnology	1603
9.56.3.20	pack_wds_SLQSGetDUNCallInfo	1603
9.56.3.21	pack_wds_SLQSGetProfileSettings	1604
9.56.3.22	pack_wds_SLQSGetRuntimeSettings	1604

9.56.3.23 pack_wds_SLQSModifyProfile . . . . .	1605
9.56.3.24 pack_wds_SLQSSet3GPPConfigItem . . . . .	1605
9.56.3.25 pack_wds_SLQSSetIPFamilyPreference . . . . .	1605
9.56.3.26 pack_wds_SLQSSetWdsEventCallback . . . . .	1606
9.56.3.27 pack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1606
9.56.3.28 pack_wds_SLQSStartDataSession . . . . .	1606
9.56.3.29 pack_wds_SLQSStopDataSession . . . . .	1607
9.56.3.30 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . . .	1607
9.56.3.31 unpack_wds_GetConnectionRate . . . . .	1608
9.56.3.32 unpack_wds_GetDefaultProfile . . . . .	1608
9.56.3.33 unpack_wds_GetDefaultProfileNum . . . . .	1608
9.56.3.34 unpack_wds_GetDormancyState . . . . .	1609
9.56.3.35 unpack_wds_GetLastMobileIPError . . . . .	1609
9.56.3.36 unpack_wds_GetMobileIP . . . . .	1609
9.56.3.37 unpack_wds_GetMobileIPProfile . . . . .	1610
9.56.3.38 unpack_wds_GetPacketStatus . . . . .	1610
9.56.3.39 unpack_wds_GetSessionDuration . . . . .	1611
9.56.3.40 unpack_wds_GetSessionState . . . . .	1611
9.56.3.41 unpack_wds_RMSetTransferStatistics . . . . .	1611
9.56.3.42 unpack_wds_SetDefaultProfile . . . . .	1612
9.56.3.43 unpack_wds_SetDefaultProfileNum . . . . .	1612
9.56.3.44 unpack_wds_SetMobileIPProfile . . . . .	1612
9.56.3.45 unpack_wds_SLQSCreateProfile . . . . .	1613
9.56.3.46 unpack_wds_SLQSDeleteProfile . . . . .	1613
9.56.3.47 unpack_wds_SLQSGet3GPPConfigItem . . . . .	1613
9.56.3.48 unpack_wds_SLQSGetCurrDataSystemStat . . . . .	1614
9.56.3.49 unpack_wds_SLQSGetDataBearerTechnology . . . . .	1614
9.56.3.50 unpack_wds_SLQSGetDUNCallInfo . . . . .	1614
9.56.3.51 unpack_wds_SLQSGetProfileSettings . . . . .	1615
9.56.3.52 unpack_wds_SLQSGetRuntimeSettings . . . . .	1615
9.56.3.53 unpack_wds_SLQSModifyProfile . . . . .	1615
9.56.3.54 unpack_wds_SLQSSet3GPPConfigItem . . . . .	1616
9.56.3.55 unpack_wds_SLQSSetIPFamilyPreference . . . . .	1616
9.56.3.56 unpack_wds_SLQSSetPacketSrvStatusCallback . . . . .	1616
9.56.3.57 unpack_wds_SLQSSetWdsEventCallback . . . . .	1616
9.56.3.58 unpack_wds_SLQSSetWdsEventCallback_ind . . . . .	1617
9.56.3.59 unpack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1617
9.56.3.60 unpack_wds_SLQSStartDataSession . . . . .	1617
9.56.3.61 unpack_wds_SLQSStopDataSession . . . . .	1618
9.56.3.62 unpack_wds_SLQSWdsSwiPDPRuntimeSettings . . . . .	1618





# 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) . . . . .	31
Wireless Data Service (WDS) . . . . .	32
Device Management Service (DMS) . . . . .	33
Network Access Service (NAS) . . . . .	34
CallBack registration (CBK) . . . . .	35
Short Message Service (SMS) . . . . .	36
Position Determination Service (PDS) . . . . .	37
Card Application Toolkit (CAT) . . . . .	38
Remote Management Service (RMS) . . . . .	39
Firmware Management Service (FMS) . . . . .	40
Open Mobile Alliance Service (OMA) . . . . .	41
Specific Absorption Rate (SAR) . . . . .	42
SWI Open Mobile Alliance Service (SWIOMA) . . . . .	43
Voice Service (VOICE) . . . . .	44
Non-service specific APIs (SWI) . . . . .	45
User Identity Module Service (UIM) . . . . .	46
Audio Service (AUDIO) . . . . .	47
Quality of Service (QOS) . . . . .	48
IMS Service (IMS) . . . . .	49
SWI Audio Service(SWIAUDIO) . . . . .	50
Location Service(LOC) . . . . .	51
Thermal Mitigation Device(TMD) . . . . .	52



## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Tables</a> . . . . .	53
----------------------------------	----



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_getIndicationRegResp</a>	55
<a href="#">_GetProfileSettingIn</a>	56
<a href="#">_GetProfileSettingOut</a>	56
<a href="#">_getResetInfoNotification</a>	57
<a href="#">_getTransLayerInfoResp</a>	58
<a href="#">_getTransNWRegInfoResp</a>	58
<a href="#">_MitigationDevInfo</a>	59
<a href="#">_modemTempNotification</a>	59
<a href="#">_packetSrvStatus</a>	60
<a href="#">_qaQmi3GPP2BroadcastCfgInfo</a>	62
<a href="#">_qaQmi3GPPBroadcastCfgInfo</a>	63
<a href="#">_setIndicationRegReq</a>	63
<a href="#">_slqs3GPPConfigItem</a>	64
<a href="#">_SlqsNas3GppNetworkRAT_</a>	66
<a href="#">_slqsNetworkScanInfo</a>	67
<a href="#">_SLQSOMADMSessionInfo</a>	68
<a href="#">_SLQSOMADMSettings</a>	71
<a href="#">_SLQSOMADMSettingsReqParams</a>	72
<a href="#">_SLQSOMADMSettingsReqParams3</a>	73
<a href="#">_SLQSSwiGetHostDevInfoParams</a>	74
<a href="#">_SLQSSwiGetOSInfoParams</a>	75
<a href="#">_SLQSSwiGetSerialNoExtParams</a>	76
<a href="#">_SLQSSwiSetHostDevInfoParams</a>	77
<a href="#">_SLQSSwiSetOSInfoParams</a>	78
<a href="#">_sysSelectPrefInfo</a>	79
<a href="#">_sysSelectPrefParams</a>	85
<a href="#">_transLayerinfo</a>	91
<a href="#">_transLayerInfoNotification</a>	91
<a href="#">_transNWRegInfoNotification</a>	92
<a href="#">accelAcceptReady_s</a>	93
<a href="#">accelTempAcceptReady_s</a>	93
<a href="#">acqOrderPref</a>	94
<a href="#">ActPilotPNElement</a>	95
<a href="#">AddCDMASysInfo</a>	95
<a href="#">AddSysInfo</a>	96
<a href="#">airTimer</a>	96
<a href="#">allCallsAlphaIDInfo</a>	97
<a href="#">allCallsDiagInfo</a>	97

<a href="#">allCallsUUSInfo</a>	98
<a href="#">alphaIDInfo</a>	98
<a href="#">altitudeSrcInfo</a>	99
<a href="#">appStats</a>	100
<a href="#">appStatus</a>	103
<a href="#">arrAlertingPattern</a>	106
<a href="#">arrAlertingType</a>	107
<a href="#">arrAlphaID</a>	108
<a href="#">arrCalledPartyNum</a>	108
<a href="#">arrCallEndReason</a>	109
<a href="#">arrCallInfo</a>	110
<a href="#">arrConnectPartyNum</a>	110
<a href="#">arrDiagInfo</a>	111
<a href="#">arrRedirPartyNum</a>	111
<a href="#">arrRemotePartyName</a>	112
<a href="#">arrRemotePartyNum</a>	112
<a href="#">arrSvcOption</a>	113
<a href="#">arrUUSInfo</a>	114
<a href="#">authenticateResult</a>	114
<a href="#">authenticationData</a>	115
<a href="#">BandCapabilityResp</a>	116
<a href="#">BdsSV</a>	119
<a href="#">BdsSVInfo</a>	119
<a href="#">BroadcastConfig</a>	120
<a href="#">burstDTMFInfo</a>	121
<a href="#">CallBarringSysInfo</a>	121
<a href="#">callBarStatus</a>	122
<a href="#">calledPartyInfo</a>	123
<a href="#">calledPartySubAdd</a>	125
<a href="#">callerIDInfo</a>	126
<a href="#">callFwdTypeAndPlan</a>	126
<a href="#">callFWExtInfo</a>	127
<a href="#">callFWInfo</a>	129
<a href="#">callInfo</a>	130
<a href="#">callingPartyInfo</a>	132
<a href="#">cardResult</a>	133
<a href="#">cardStatus</a>	134
<a href="#">CarrierImage_t</a>	135
<a href="#">CatAIPhalIdentifierTlv</a>	136
<a href="#">CatCommonEventTlv</a>	137
<a href="#">CatEndProactiveSessionTlv</a>	137
<a href="#">CATEventDataType</a>	138
<a href="#">CatEventIDDDataTlv</a>	138
<a href="#">CatEventListTlv</a>	138
<a href="#">CatRefreshTlv</a>	139
<a href="#">ccSUPSType</a>	139
<a href="#">CDMABroadcastConfig</a>	140
<a href="#">CDMAChannel</a>	141
<a href="#">CDMAECIOThresh</a>	142
<a href="#">CDMAInfo</a>	142
<a href="#">cdmaMsgDecodingParams</a>	143
<a href="#">cdmaMsgEncodingParams</a>	146
<a href="#">CDMARSSIThresh</a>	148
<a href="#">CDMASSInfo</a>	148
<a href="#">cdmaSSInfo</a>	149
<a href="#">CDMASysInfo</a>	149
<a href="#">CDMASysInfoExt</a>	153
<a href="#">CellDb</a>	153

cellParams	154
changeUIMPIN	155
ChannelRate	156
channelRate	157
CLIPResp	157
CLIRResp	158
ClkInfo	159
CNAPResp	160
COLPResp	160
COLRResp	161
CommInfo	162
ConnectionStatus	163
connectionStatus	164
connectNumInfo	164
CrashInfo	166
CrashInfoParams	167
CreateProfileIn	168
CreateProfileOut	169
CSGID	169
CUGInfo	170
curAMRConfig	171
CurrDataSysStat	172
currentCatEvent	172
CurrentImgList	173
currentPLMN	174
CurrImageInfo	175
CurrNetworkInfo	176
currNetworkInfo	178
custFeaturesInfo	179
custFeaturesSetting	181
custSettingInfo	183
custSettingList	184
dataBearers	184
DataBearerTech	185
DataBearerTechExt	187
dataBearerTechnology	188
dataRate	189
dataSrvCapabilities	190
DataStatusDetail	191
DataULongLongTlv	193
DataULongTlv	193
DcsUsbPortNames	193
delAssistDataStatus	193
depersonalizationInformation	194
detailSvcInfo	195
DeviceConfigDetail	197
DHCPOption	198
DHCPOptionList	198
diagInfo	199
dirNum	199
dms_ActivationStatusTlv	200
dms_OperatingModeTlv	201
dmsCurrentPRLInfo	201
DMScustSettingInfo	202
DMScustSettingList	203
DMSgetCustomFeatureV2	203
DMSgetCustomInput	204
dmsIndicationRegisterReq	204

dmsSwiGetResetInfo	205
Domain	206
DomainNameList	206
DRCParams	207
DTMFInfo	207
DTMFLengths	208
DUNCallInfoInd	209
dunchannelRate	210
ecioListElement	210
ECIOThresh	211
ECTNum	212
encryptedPIN1	212
ERIFileparams	213
errorRateListElement	213
eTWSPLMNInfoTlv	214
extDispRecInfo	215
FactorySequenceNumber	215
fileAttributes	216
fileInfo	219
FirmwareUpdatStat	220
FMSImageElement	222
FMSImageIdElement	222
FMSImageIdEntries	223
FMSImageList	224
FMSPrefImageList	225
fwinfo_s	225
GERANInfo	226
geranInstInfo	228
getAllCallInformation	229
getAllCallRmtPtyName	229
getAllCallRmtPtyNum	230
GetAudioPathConfigReq	230
GetAudioPathConfigResp	231
GetAudioProfileReq	233
GetAudioProfileResp	233
GetAudioVolTLBConfigReq	235
GetAudioVolTLBConfigResp	235
getCallFWExtInfo	236
getCallFWInfo	236
getCustomFeatureV2	237
getCustomInput	238
getDUNCallInfoReq	238
getDUNCallInfoResp	240
getDyingGaspCfg	242
getDyingGaspStatistics	243
GetErrRateResp	244
GetHRPDStatsResp	244
GetIMSSMSConfigParams	245
GetIMSUserConfigParams	246
GetIMSVoIPConfigResp	247
GetInstIDResp	249
GetM2MAudioProfileReq	250
GetM2MAudioProfileResp	250
GetM2MAudioVolumeReq	251
GetM2MAudioVolumeResp	252
GetM2MAVMuteReq	252
GetM2MAVMuteResp	253
GetM2MSpkrGainReq	253

GetM2MSpkrGainResp	254
getMsgWaitingInfo	254
GetNetworkTimeResp	255
GetRegMgrConfigParams	255
GetSessionIDResp	256
GetSIPConfigResp	256
GnssData	258
gnssSvInfoNotification	259
GPRSQoS	260
GPRSRequestedQoS	261
GPSSStateInfo	262
gpsTime_s	266
gsmCellInfo	266
GSMRSSIThresh	268
GSMSrvStatusInfo	268
GSMSysInfo	269
gyroAcceptReady_s	272
gyroTempAcceptReady_s	273
HDRECIOThresh	273
HDRIOThresh	274
HDRPersonalityInd	274
HDRPersonalityResp	275
HDRProtSubtypResp	275
HDRRSSIThresh	276
HDRSINRThresh	277
HDRSINRThreshold	277
HDRSSInfo	278
hdrSSInfo	279
HDRSysInfo	280
homeSIDNID	282
hotSwapStatus	282
image_info_t	283
ImageElement	283
ImageIdElement	284
ImageIDEntries	285
ImageList	286
IMSAIndRegisterInfo	286
imsaPdpStatusInfo	287
imsaRatStatusInfo	288
IMSARegistrationStatus	289
imsaRegStatusInfo	290
IMSAServiceStatus	290
IMSASupportedFieldsResp	292
IMSASupportedMsgInfo	293
imsaSvcStatusInfo	293
imsCfgIndRegisterInfo	294
imsRegMgrConfigInfo	296
imsSIPConfigInfo	296
imsSMSConfigInfo	297
imsUserConfigInfo	298
imsVoIPConfigInfo	299
IndFieldsList	301
infoInterFreq	302
IOThresh	303
IPv4Addr	303
IPv6Addr	304
IPV6AddressInfo	304
ipv6AddressInfo	305

IPV6GWAddressInfo	305
IPv6TrafCls	306
LibPackGPRSRequestedQoS	306
LibpackProfile3GPP	307
LibpackProfile3GPP2	313
LibPackprofile_3GPP	318
LibPackprofile_3GPP2	324
LibPackQosClassID	329
LibPackTFTIDParams	330
LibPackUMTSQoS	332
LibPackUMTSReqQoSSigInd	334
lineCtrlInfo	335
loc_BdsSV	336
loc_BdsSVInfo	336
loc_CellDb	337
loc_ClkInfo	337
loc_GnssData	339
loc_gpsTime	340
loc_LocApplicationInfo	341
loc_precisionDilution	342
loc_sensorDataUsage	343
loc_SV	344
loc_SVInfo	344
loc_svUsedforFix	345
LocApplicationInfo	346
LocDelAssDataReq	347
LOCEventRegisterReqResp	348
LOCExtPowerStateReqResp	350
LocInjectPositionReq	350
LocInjectSensorDataReq	354
LocSetCradleMountReq	355
LOCStartReq	356
LOCStopReq	358
LteCQIParm	358
LteEARFCN	359
LteGsmCellInfo	360
LTEInfo	361
LTEInfoInterfreq	363
LTEInfoIntrafreq	363
LTEInfoNeighboringGSM	366
LTEInfoNeighboringWCDMA	366
LteNasReleaseInfo_s	367
LtePCI	368
LteRsrpinformation	368
LTERSRPThresh	369
LTERSRQThresh	369
LTERSSIThresh	370
LTESigRptCfg	370
LTESigRptConfig	371
LteSnrinformation	372
LTESNRThresh	373
LTESNRThreshold	373
LTESSInfo	374
LteSSInfo	375
LTESysInfo	375
LteWcdmaCellInfo	377
messageModeTlv	379
messageWaitingInfoContent	379

minBasedIMSI	380
mitigationDevList	381
MNRInfo	381
ModifyProfileIn	382
ModifyProfileOut	383
msgWaitingInfo	383
namName	384
nas_acqOrderPref	384
nas_AddCDMASysInfo	385
nas_AddSysInfo	385
nas_CallBarringSysInfo	386
nas_callBarStatus	387
nas_CDMAECIOThresh	388
nas_CDMAInfo	388
nas_CDMARSSIThresh	389
nas_CDMASysInfo	390
nas_CDMASysInfoExt	393
nas_cellParams	394
nas_CommInfo	395
nas_CSGID	397
nas_currentPLMN	398
nas_dataSrvCapabilities	399
nas_detailSvcInfo	399
nas_ecioListElement	401
nas_errorRateListElement	402
nas_GERANInfo	402
nas_geranInstInfo	404
nas_gsmCellInfo	405
nas_GSMRSSIThresh	406
nas_GSMSrvStatusInfo	407
nas_GSMSysInfo	408
nas_HDRECIOThresh	411
nas_HDRIOThresh	411
nas_HDRRSSIThresh	412
nas_HDRSINRThreshold	412
nas_HDRSysInfo	413
nas_infoInterFreq	415
nas_lteGsmCellInfo	416
nas_LTEInfo	417
nas_LTEInfoInterfreq	420
nas_LTEInfoIntrafreq	420
nas_LTEInfoNeighboringGSM	422
nas_LTEInfoNeighboringWCDMA	423
nas_lteRsrpInformation	424
nas_LTERSRRPThresh	424
nas_LTERSRRQThresh	425
nas_LTERSSIThresh	425
nas_LTESigRptConfig	426
nas_lteSnrinformation	426
nas_LTESNRThreshold	427
nas_LTESysInfo	427
nas_lteWcdmaCellInfo	430
nas_MNRInfo	431
nas_netSelectionPref	432
nas_nmrCellInfo	432
nas_PhyCaAggPcellInfo	434
nas_PhyCaAggScellDIBw	434
nas_PhyCaAggScellIndex	435

<a href="#">nas_PhyCaAggScellIndType</a>	435
<a href="#">nas_PhyCaAggScellInfo</a>	436
<a href="#">nas_qaQmi3Gpp2TimeZone</a>	439
<a href="#">nas_QmiNas3GppNetworkInfo</a>	440
<a href="#">nas_QmiNas3GppNetworkRAT</a>	440
<a href="#">nas_QmisNasPcsDigit</a>	441
<a href="#">nas_RejectReasonTlv</a>	442
<a href="#">nas_RFInfoTlv</a>	442
<a href="#">nas_roamIndList</a>	443
<a href="#">nas_rsrqInformation</a>	444
<a href="#">nas_rxSignalStrengthListElement</a>	444
<a href="#">nas_servSystem</a>	445
<a href="#">nas_SignalStrengthTlv</a>	447
<a href="#">nas_SLQSSignalStrengthsIndReq</a>	447
<a href="#">nas_SLQSSignalStrengthsInformation</a>	448
<a href="#">nas_SLQSSignalStrengthsTlv</a>	449
<a href="#">nas_SrvStatusInfo</a>	449
<a href="#">nas_sysInfoCommon</a>	450
<a href="#">nas_TDSCDMAECIOThresh</a>	453
<a href="#">nas_TDSCDMARSCPTthresh</a>	453
<a href="#">nas_TDSCDMARSSIThresh</a>	454
<a href="#">nas_TDSCDMASINRThresh</a>	454
<a href="#">nas_timeInfo</a>	455
<a href="#">nas_UMTSInfo</a>	456
<a href="#">nas_UMTSinstInfo</a>	458
<a href="#">nas_umtsLTENbrCell</a>	459
<a href="#">nas_UniversalTime</a>	460
<a href="#">nas_wcdmaCellInfo</a>	461
<a href="#">nas_WCDMAECIOThresh</a>	462
<a href="#">nas_WCDMAInfoLTENeighborCell</a>	463
<a href="#">nas_WCDMARSSIThresh</a>	463
<a href="#">nas_WCDMASysInfo</a>	464
<a href="#">NASBandPreferenceTlv</a>	467
<a href="#">nasCellLocationInfoResp</a>	468
<a href="#">NASEmergencyModeTlv</a>	469
<a href="#">nasGet3GPP2SubscriptionInfoReq</a>	469
<a href="#">nasGet3GPP2SubscriptionInfoResp</a>	470
<a href="#">nasGetHDRColorCodeResp</a>	470
<a href="#">nasGetLTECphyCa</a>	471
<a href="#">NasGetLTECphyCaInfo</a>	471
<a href="#">nasGetLTECphyCaResp</a>	472
<a href="#">nasGetSigInfoResp</a>	472
<a href="#">nasGetSysInfoResp</a>	473
<a href="#">nasGetTxRxInfoReq</a>	476
<a href="#">nasGetTxRxInfoResp</a>	477
<a href="#">NASGWAcqOrderPrefTlv</a>	477
<a href="#">nasIndicationRegisterReq</a>	477
<a href="#">nasInitNetworkReg</a>	480
<a href="#">NASLTEBandPreferenceTlv</a>	481
<a href="#">NASLteNasReleaseInfoTlv</a>	481
<a href="#">NASModePreferenceTlv</a>	482
<a href="#">NASNetSelPreferenceTlv</a>	482
<a href="#">nasNetworkTime</a>	482
<a href="#">nasOperatorNameResp</a>	483
<a href="#">NASOTAMessageTlv</a>	484
<a href="#">NASPhyCaAggPcellInfo</a>	484
<a href="#">NASPhyCaAggScellIDIBw</a>	485
<a href="#">NASPhyCaAggScellIndex</a>	486

NASPhyCaAggScellIndType	486
NASPhyCaAggScellInfo	487
nasPLMNNameReq	488
nasPLMNNameResp	489
NASPRLPreferenceTlv	491
NASQmiCbkNasSwiOTAMessageInd	492
NASQmiCbkNasSystemSelPrefInd	492
NASRoamPreferenceTlv	493
NASServDomainPrefTlv	493
NASServingSystemInfo	493
nasSigInfo	495
nasSwiGetChannelLockResp	496
NasSwiIndReg	496
nasSwiSetChannelLockReq	498
nasSysInfo	498
NASTimeInfoTlv	501
netSelectionPref	501
NetStats	502
NetworkDebugResp	503
NetworkStat1x	504
NetworkStatEVDO	506
newMTMessageTlv	508
newPwdData	508
nmrCellInfo	509
NSSAudioCtrl	510
NWProfile	511
omaDmConfigTlv	511
omaDmConfigTlvExt	512
omaDmFotaTlv	514
omaDmFotaTlvExt	516
omaDmNotificationsTlv	518
operatorNameString	518
OperatorPLMNData	519
operatorPLMNList	520
pack_dms_GetCustFeaturesV2_t	520
pack_dms_SetCrashAction_t	521
pack_dms_SetCustFeature_t	521
pack_dms_SetCustFeaturesV2_t	522
pack_dms_SetEventReport_t	523
pack_dms_SetPower_t	523
pack_dms_SetUSBComp_t	523
pack_dms_SLQSDmsSwiIndicationRegister_t	523
pack_dms_SLQSSwiSetDyingGaspCfg_t	524
pack_dms_UIMGetICCID_t	524
pack_fms_GetImagesPreference_t	525
pack_fms_GetStoredImages_t	525
pack_fms_SetImagesPreference_t	525
pack_loc_Delete_Assist_Data_t	526
pack_loc_EventRegister_t	527
pack_loc_SetExtPowerState_t	529
pack_loc_SetOperationMode_t	529
pack_loc_SLQSLOCGetBestAvailPos_t	530
pack_loc_Start_t	530
pack_loc_Stop_t	532
pack_nas_SetACCOLC_t	532
pack_nas_SetNetworkPreference_t	533
pack_nas_SLQSGetPLMNName_t	534
pack_nas_SLQSInitiateNetworkRegistration_t	534

pack_nas_SLQSNasConfigSigInfo2_t	535
pack_nas_SLQSNasIndicationRegisterExt_t	540
pack_nas_SLQSNasSwiOTAMessageCallback_t	543
pack_nas_SLQSSetSignalStrengthsCallback_t	544
pack_nas_SLQSSetSysSelectionPref_t	544
pack_qmi_t	549
pack_qos_SLQSQosSwiReadApnExtraParams_t	549
pack_qos_SLQSQosSwiReadDataStats_t	550
pack_qos_SLQSSetQosEventCallback_t	550
pack_sms_SendSMS_t	551
pack_sms_SetNewSMSCallback_t	551
pack_sms_SLQSDeleteSMS_t	552
pack_sms_SLQSGetSMS_t	553
pack_sms_SLQSGetSMSList_t	553
pack_sms_SLQSModifySMSStatus_t	554
pack_swiloc_SwiLocSetAutoStart_t	555
pack_swisma_SLQSOMADMCancelSession_t	556
pack_swisma_SLQSOMADMGetSessionInfo_t	557
pack_swisma_SLQSOMADMSelectSelection_t	557
pack_swisma_SLQSOMADMSetSettings_t	558
pack_swisma_SLQSOMADMStartSession_t	559
pack_uim_ChangePin_t	560
pack_uim_ReadTransparent_t	561
pack_uim_SetPinProtection_t	562
pack_uim_SLQSUIEventRegister_t	563
pack_uim_SLQSUIMSwitchSlot_t	563
pack_uim_UnblockPin_t	564
pack_uim_VerifyPin_t	565
pack_wds_GetDefaultProfile_t	566
pack_wds_GetDefaultProfileNum_t	566
pack_wds_GetDormancyState_t	567
pack_wds_GetLastMobileIPError_t	567
pack_wds_GetMobileIP_t	567
pack_wds_GetMobileIPProfile_t	567
pack_wds_GetPacketStatus_t	567
pack_wds_GetSessionDuration_t	568
pack_wds_RMSetTransferStatistics_t	568
pack_wds_SetDefaultProfile_t	568
pack_wds_SetDefaultProfileNum_t	569
pack_wds_SetMobileIPProfile_t	569
pack_wds_SLQSCreateProfile_t	571
pack_wds_SLQSDeleteProfile_t	571
pack_wds_SLQSGetCurrDataSystemStat_t	572
pack_wds_SLQSGetDataBearerTechnology_t	572
pack_wds_SLQSGetDUNCallInfo_t	572
pack_wds_SLQSGetProfileSettings_t	573
pack_wds_SLQSGetRuntimeSettings_t	573
pack_wds_SLQSModifyProfile_t	574
pack_wds_SLQSSet3GPPConfigItem_t	575
pack_wds_SLQSSetIPFamilyPreference_t	576
pack_wds_SLQSSetWdsEventCallback_t	576
pack_wds_SLQSSetDHCPv4ClientConfig_t	577
pack_wds_SLQSStartDataSession_t	577
pack_wds_SLQSStopDataSession_t	579
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	579
PackCreateProfileOut	579
packgetDyingGaspCfg	580
packgetDyingGaspStatistics	580

PCMparams	581
PCSCFFQDNAddress	581
PCSCFFQDNAddressList	582
PCSCFIPv4ServerAddressList	582
PDSPositionData	583
PDSPosMethodStateReq	585
peerNumberInfo	585
personalizationStatus	587
PhyCaAggPcellInfo	588
PhyCaAggScellIDBw	589
PhyCaAggScellIndex	590
PhyCaAggScellIndType	590
PhyCaAggScellInfo	591
PilotSetData	593
PilotSetParams	594
pktErrRate	594
PLMNNetworkName	595
PLMNNetworkNameData	595
Port	597
precisionDilution_s	597
PrefImageList	598
prefVoiceSO	598
Profile3GPP	600
Profile3GPP2	606
ProfileIdentifier	611
protocolSubtypeElement	612
PSDetachReq	614
qaQmi3Gpp2TimeZone	614
qaQmiInterfaceInfo	615
qaQmiServingSystemParam	615
QmiCbkCatEventStatusReportInd	619
QmiCbkLocBestAvailPosInd	620
QmiCbkLocCradleMountInd	625
QmiCbkLocEngineStateInd	626
QmiCbkLocEventTimeSyncInd	626
QmiCbkLocInjectPositionInd	627
QmiCbkLocInjectSensorDataInd	628
QmiCbkLocInjectTimeInd	629
QmiCbkLocInjectUTCTimeInd	629
QmiCbkLocPositionReportInd	630
QmiCbkLocSensorStreamingInd	636
QmiCbkLocSetExtPowerConfigInd	636
QmiCbkNasLTECphyCalInfo	637
QmiCbkSwiOmaDmEventStatusReportInd	638
QmiCbkSwiOmaDmEventStatusReportIndExt	638
QmiCbkTmdMitiLvlRptInd	638
QmiCbkWdsStatisticsIndState	639
qmifwinfo_s	640
QmiNas3GppNetworkInfo	641
QmiNasGetRFBandInfoResp	642
QmiNasPerformNetworkScanResp	642
qmiSmsMessageList	643
qmiWSDDataBearerTechnology	643
QmiWdsIpAddressInfo	644
qmiWdsRunTimeSettings	644
QosClassID	648
QosEventInfo	649
QosFlowInfo	651

QosFlowInfoState	652
QosMap	652
RankIndicatorInd	653
readResult	653
readTransparentInfo	654
redirNumInfo	654
registerRefresh	656
remainingRetries	657
remotePartyName	657
remotePartyNum	658
ReqFieldsList	659
RespFieldsList	660
RFBandInfoElements	660
rmTrasferStaticsReq	661
roamIndList	661
RoamingInfo	662
roamTimer	662
RSRPThresh	663
rsrqInformation	664
RSRQThresh	664
RSSIThresh	665
RXAGCList	666
RXAVCList	667
rxInfo	667
RXPCMIIRFtr	669
rxSignalStrengthListElement	670
sApnExtraParams	671
satelliteInfo	672
sensorData	674
sensorDataUsage_s	676
serialNumbersInfo	677
serviceProviderName	678
ServingSystemInfo	678
servSystem	680
sessionInfo	681
sessionInfoExt	682
sessionInfoTlv	682
sessionInfoTlvExt	683
SetAudioPathConfigReq	683
SetAudioProfileReq	685
SetAudioVoITLBConfigReq	686
SetAudioVoITLBConfigResp	687
setCustomSettingV2	688
setDyingGaspCfg	688
SetIMSSMSCConfigReq	689
SetIMSSMSCConfigResp	690
SetIMSUserConfigReq	690
SetIMSUserConfigResp	691
SetIMSVoIPConfigReq	691
SetIMSVoIPConfigResp	693
SetM2MAudioAVCFGReq	694
SetM2MAudioLPBKReq	695
SetM2MAudioProfileReq	695
SetM2MAudioVolumeReq	696
SetM2MAVMuteReq	697
SetM2MSprkGainReq	698
setPINProtection	698
SetRegMgrConfigReq	699

SetRegMgrConfigResp	700
setSignalStrengthInfo	700
SetSIPConfigReq	705
SetSIPConfigResp	706
sGetDeviceSeriesResult	707
sidNid	707
sigInfo	708
signalInfo	709
SignalStrengthDataType	710
slot_t	710
slotInf	711
slotInfo	713
slots_t	714
slqsautoconnect	714
SLQSDDeleteProfileParams	715
slqsfwinfo_s	716
SlqsNas3GppNetworkInfo	717
SlqsNasPcsDigit	718
slqssendasyncsmsparams_s	719
slqssendsmsparams_s	721
slqsSessionStateInfo	722
slqsSignalStrengthInfo	723
SLQSSignalStrengthsIndReq	726
SLQSSignalStrengthsInformation	728
slqsWdsEventInfo	729
SMSAsyncRawSend_s	731
sMSCAddress	732
SMSCAddress	733
sMSCAddressTlv	733
sMSEtwsMessage	734
SMSEtwsMessage	734
sMSEtwsMessageTlv	735
sMSEtwsPlmn	735
SMSEtwsPlmn	736
SMSEventInfo_s	736
smsMaxStorageSizeReq	738
smsMaxStorageSizeResp	738
SMSMemoryInfo	739
sMSMessageMode	739
SMSMessageMode	740
smsMsgprotocolResp	740
sMSMTMessage	741
SMSMTMessage	741
sMSOnIMS	742
SMSOnIMS	742
sMSOnIMSTlv	742
smsRouteEntry	743
smsSetRoutesReq	744
sMSTransferRouteMTMessage	745
SMSTransferRouteMTMessage	745
sQosFlowStat	746
sQosStat	747
SrvStatusInfo	748
ssdatasession_params	749
SupportedMsgList	752
SUPInfo	752
SV	753
SVInfo	754

svUsedforFix_s	755
SWI_STRUCT_CarrierImage	755
SwiLocGetAutoStartResp	756
SwiLocSetAutoStartReq	758
swiModemStatusResp	760
SwiOTAMsg_s	760
swiPDPRuntimeSettingsReq	761
swiPDPRuntimeSettingsResp	762
swiQosFilter	764
swiQosFlow	767
swiQosGranted	770
swiQosIds	770
swiQosModifyReq	771
swiQosReq	771
swiRMTrasnferStaticsReq	772
sysInfoCommon	773
t_gpsTime	775
t_sensor	775
t_Sv	776
TDSCDMAECIOThresh	776
TDSCDMARSCPTthresh	776
TDSCDMARSSIThresh	777
TDSCDMASigInfoExt	777
tdscdmaSigInfoExt	778
TDSCDMASINRCONFTthresh	779
TDSCDMASINRThresh	779
tempratureData	780
TFTIDParams	781
timeInfo	783
TmdDeRegNotMitigationLvlReq	784
TmdGetMitigationDevListResp	785
TmdGetMitigationLvlReq	785
TmdGetMitigationLvlResp	786
TmdMitigationLvlIndReq	787
TmdRegNotMitigationLvlReq	787
tokenBucket	788
Tos	788
transferRouteMessageTlv	789
TransferStatInd	789
transferStatInd	790
TransferStatsDataType	790
TrStatInd	790
trueIMSI	791
TXAGCList	792
txInfo	793
TXPCMIIRFtr	794
uim_appStatus	795
uim_cardResult	798
uim_cardStatus	799
uim_changeUIMPIN	800
uim_encryptedPIN1	801
uim_fileInfo	801
uim_hotSwapStatus	802
uim_readResult	802
uim_readTransparentInfo	803
uim_remainingRetries	804
uim_sessionInformation	804
uim_setPINProtection	805

uim_slotInfo	806
uim_UIMSessionInformation	808
uim_unblockUIMPIN	808
uim_verifyUIMPIN	809
UIMAuthenticateReq	810
UIMAuthenticateResp	811
UIMChangePinReq	812
UIMDepersonalizationReq	812
UIMDepersonalizationResp	813
UIMEventRegisterReqResp	813
UIMGetCardStatusResp	814
UIMGetConfigurationReq	814
UIMGetConfigurationResp	815
UIMGetFileAttributesReq	816
UIMGetFileAttributesResp	816
UIMGetSlotsStatusResp	817
UIMPinResp	818
UIMPowerDownReq	818
UIMPowerUpReq	819
UIMReadTransparentReq	819
UIMReadTransparentResp	820
UIMRefreshCompleteReq	821
UIMRefreshEvent	822
UIMRefreshGetLastEventReq	823
UIMRefreshGetLastEventResp	824
UIMRefreshOKReq	824
UIMRefreshRegisterReq	825
UIMSessionInformation	825
UIMSetPinProtectionReq	826
UIMSlotsStatus	827
UIMSlotStatus	827
UIMSlotStatusChangeInfo	829
UIMStatusChangeInfo	829
UIMSwitchSlotReq	829
UIMUnblockPinReq	830
UIMVerifyPinReq	831
UMTSInfo	832
UMTSInstInfo	834
umtsLTENbrCell	834
UMTSMInQoS	835
UMTSQoS	838
UMTSReqQoSsigInd	840
unblockUIMPIN	841
UniversalTime	842
unpack_dms_GetActivationState_t	843
unpack_dms_GetBandCapability_t	844
unpack_dms_GetCrashAction_t	844
unpack_dms_GetCustFeature_t	844
unpack_dms_GetCustFeaturesV2_t	845
unpack_dms_GetDeviceCap_t	846
unpack_dms_GetDeviceCapabilities_t	846
unpack_dms_GetDeviceHardwareRev_t	847
unpack_dms_GetDeviceMfr_t	847
unpack_dms_GetDeviceSerialNumbers_t	847
unpack_dms_GetFirmwareInfo_t	848
unpack_dms_GetFirmwareRevision_t	849
unpack_dms_GetFirmwareRevisions_t	849
unpack_dms_GetFSN_t	850

<a href="#">unpack_dms_GetHardwareRevision_t</a>	850
<a href="#">unpack_dms_GetIMSI_t</a>	851
<a href="#">unpack_dms_GetModelID_t</a>	851
<a href="#">unpack_dms_GetNetworkTime_t</a>	851
<a href="#">unpack_dms_GetPower_t</a>	852
<a href="#">unpack_dms_GetPRLVersion_t</a>	853
<a href="#">unpack_dms_GetSerialNumbers_t</a>	853
<a href="#">unpack_dms_GetUSBComp_t</a>	853
<a href="#">unpack_dms_GetVoiceNumber_t</a>	854
<a href="#">unpack_dms_SetCrashAction_t</a>	854
<a href="#">unpack_dms_SetCustFeature_t</a>	854
<a href="#">unpack_dms_SetCustFeaturesV2_t</a>	855
<a href="#">unpack_dms_SetEventReport_ind_t</a>	855
<a href="#">unpack_dms_SetEventReport_t</a>	856
<a href="#">unpack_dms_SetFirmwarePreference_t</a>	856
<a href="#">unpack_dms_SetPower_t</a>	856
<a href="#">unpack_dms_SetUSBComp_t</a>	856
<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t</a>	856
<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_t</a>	857
<a href="#">unpack_dms_SLQSDmsSwiIndicationRegister_t</a>	858
<a href="#">unpack_dms_SLQSGetBandCapability_t</a>	859
<a href="#">unpack_dms_SLQSSwiClearDyingGaspStatistics_t</a>	862
<a href="#">unpack_dms_SLQSSwiGetDyingGaspCfg_t</a>	862
<a href="#">unpack_dms_SLQSSwiGetDyingGaspStatistics_t</a>	863
<a href="#">unpack_dms_SLQSSwiGetFirmwareCurr_t</a>	863
<a href="#">unpack_dms_SLQSSwiGetFwUpdateStatus_t</a>	864
<a href="#">unpack_dms_SLQSSwiSetDyingGaspCfg_t</a>	866
<a href="#">unpack_dms_UIMGetICCID_t</a>	866
<a href="#">unpack_fms_GetImagesPreference_t</a>	867
<a href="#">unpack_fms_GetStoredImages_t</a>	867
<a href="#">unpack_fms_SetImagesPreference_t</a>	868
<a href="#">unpack_loc_BestAvailPos_Ind_t</a>	868
<a href="#">unpack_loc_Delete_Assist_Data_t</a>	874
<a href="#">unpack_loc_EngineState_Ind_t</a>	875
<a href="#">unpack_loc_EventRegister_t</a>	875
<a href="#">unpack_loc_PositionRpt_Ind_t</a>	876
<a href="#">unpack_loc_SetExtPowerConfig_Ind_t</a>	882
<a href="#">unpack_loc_SetExtPowerState_t</a>	882
<a href="#">unpack_loc_SetOperationMode_t</a>	883
<a href="#">unpack_loc_SLQSLOCGetBestAvailPos_t</a>	883
<a href="#">unpack_loc_Start_t</a>	883
<a href="#">unpack_loc_Stop_t</a>	884
<a href="#">unpack_nas_GetCDMANetworkParameters_t</a>	884
<a href="#">unpack_nas_GetHomeNetwork_t</a>	885
<a href="#">unpack_nas_GetNetworkPreference_t</a>	886
<a href="#">unpack_nas_GetRFInfo_t</a>	887
<a href="#">unpack_nas_GetServingNetwork_t</a>	887
<a href="#">unpack_nas_GetServingNetworkCapabilities_t</a>	888
<a href="#">unpack_nas_GetSignalStrengths_t</a>	889
<a href="#">unpack_nas_PerformNetworkScan_t</a>	889
<a href="#">unpack_nas_SetDataCapabilitiesCallback_ind_t</a>	890
<a href="#">unpack_nas_SetEventReportInd_t</a>	890
<a href="#">unpack_nas_SetNasLTECphyCaIndCallback_ind_t</a>	891
<a href="#">unpack_nas_SetNetworkPreference_t</a>	892
<a href="#">unpack_nas_SetRoamingIndicatorCallback_ind_t</a>	893
<a href="#">unpack_nas_SetServingSystemCallback_ind_t</a>	893
<a href="#">unpack_nas_SlqsGetLTECphyCAInfo_t</a>	893
<a href="#">unpack_nas_SLQSGetNetworkTime_t</a>	894

unpack_nas_SLQSGetPLMNName_t	894
unpack_nas_SLQSGetServingSystem_t	895
unpack_nas_SLQSGetSignalStrength_t	897
unpack_nas_SLQSGetSysInfo_t	899
unpack_nas_SLQSGetSysSelectionPref_t	901
unpack_nas_SLQSNasGetCellLocationInfo_t	905
unpack_nas_SLQSNasGetSigInfo_t	907
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t	907
unpack_nas_SLQSNasSigInfoCallback_ind_t	908
unpack_nas_SLQSNasSwiModemStatus_t	909
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t	910
unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t	910
unpack_nas_SLQSSwiGetLteCQI_t	910
unpack_nas_SLQSSysInfoCallback_ind_t	911
unpack_omaDmConfigTlv_t	914
unpack_omaDmFotaTlv_t	915
unpack_omaDmNotificationsTlv_t	917
unpack_qmi_t	917
unpack_qos_dataRate_t	918
unpack_qos_IPv4Addr_t	918
unpack_qos_IPv6Addr_t	919
unpack_qos_IPv6TrafCls_t	919
unpack_qos_pktErrRate_t	920
unpack_qos_Port_t	920
unpack_qos_QosFlowInfo_t	921
unpack_qos_QosFlowInfoState_t	922
unpack_qos_SLQSQosGetNetworkStatus_t	923
unpack_qos_SLQSQosSwiReadApnExtraParams_t	923
unpack_qos_SLQSQosSwiReadDataStats_t	925
unpack_qos_SLQSSetQosEventCallback_ind_t	926
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	926
unpack_qos_SLQSSetQosPriEventCallback_ind_t	927
unpack_qos_SLQSSetQosStatusCallback_ind_t	927
unpack_qos_swiQosFilter_t	929
unpack_qos_swiQosFlow_t	932
unpack_qos_tokenBucket_t	936
unpack_qos_Tos_t	937
unpack_QosFlowStat_t	937
unpack_sms_SendSMS_t	938
unpack_sms_SetNewSMSCallback_ind_t	938
unpack_sms_SetNewSMSCallback_t	939
unpack_sms_SLQSDeleteSMS_t	940
unpack_sms_SLQSGetSMS_t	940
unpack_sms_SLQSGetSMSList_t	940
unpack_sms_SLQSModifySMSStatus_t	941
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	941
unpack_swiloc_SwiLocGetAutoStart_t	942
unpack_swioma_SLQSOMADMAAlertCallback_ind_t	943
unpack_swioma_SLQSOMADMGetSessionInfo_t	944
unpack_swioma_SLQSOMADMGetSettings_t	947
unpack_swioma_SLQSOMADMStartSession_t	948
unpack_uim_ChangePin_t	949
unpack_uim_GetCardStatus_t	949
unpack_uim_ReadTransparent_t	950
unpack_uim_SetPinProtection_t	951
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	952
unpack_uim_SLQSUIEventRegister_t	952
unpack_uim_SLQSUIGetSlotsStatus_t	952

unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t	953
unpack_uim_UnblockPin_t	953
unpack_uim_VerifyPin_t	954
unpack_wds_GetConnectionRate_t	955
unpack_wds_GetDefaultProfile_t	955
unpack_wds_GetDefaultProfileNum_t	957
unpack_wds_GetDormancyState_t	957
unpack_wds_GetLastMobileIPError_t	957
unpack_wds_GetMobileIP_t	958
unpack_wds_GetMobileIPProfile_t	958
unpack_wds_GetPacketStatus_t	959
unpack_wds_GetSessionDuration_t	960
unpack_wds_GetSessionState_t	960
unpack_wds_RMSetTransferStatistics_t	961
unpack_wds_SetMobileIPProfile_t	961
unpack_wds_SLQSCreateProfile_t	961
unpack_wds_SLQSDeleteProfile_t	961
unpack_wds_SLQSGet3GPPConfigItem_t	962
unpack_wds_SLQSGetCurrDataSystemStat_t	963
unpack_wds_SLQSGetDataBearerTechnology_t	963
unpack_wds_SLQSGetDUNCallInfo_t	964
unpack_wds_SLQSGetProfileSettings_t	965
unpack_wds_SLQSGetRuntimeSettings_t	965
unpack_wds_SLQSModifyProfile_t	967
unpack_wds_SLQSSetIPFamilyPreference_t	967
unpack_wds_SLQSSetPacketSrvStatusCallback_t	968
unpack_wds_SLQSSetWdsEventCallback_ind_t	969
unpack_wds_SLQSSetDHCPv4ClientConfig_t	970
unpack_wds_SLQSStartDataSession_t	971
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	971
UnPackGetProfileSettingOut	973
unpackWdsProfileParam	973
USBCompConfig	973
USBCompParams	974
USSDNoWaitIndicationInfo	976
USSDRespFNetwork	977
USSInfo	977
USSResp	978
UUSInfo	979
verifyUIMPIN	980
voiceALSSelectLineInfo	980
voiceALSSetLineSwitchInfo	981
voiceAnswerCall	981
voiceBindSubscriptionInfo	982
voiceBurstDTMFInfo	982
voiceCallInfoReq	983
voiceCallInfoResp	983
voiceCallRequestParams	986
voiceCallResponseParams	987
voiceContDTMFInfo	988
voiceDTMFEventInfo	989
voiceFlashInfo	990
voiceGetAllCallInfo	990
voiceGetCallBarringReq	993
voiceGetCallBarringResp	994
voiceGetCallFWReq	995
voiceGetCallFWResp	996
voiceGetCallWaitInfo	997

voiceGetCLIPResp	999
voiceGetCLIRResp	1000
voiceGetCNAPResp	1001
voiceGetCOLPResp	1002
voiceGetCOLRResp	1004
voiceGetConfigReq	1005
voiceGetConfigResp	1007
voiceIndicationRegisterInfo	1008
voiceInfoRec	1009
voiceManageCallsReq	1012
voiceManageCallsResp	1012
voiceOrigUSSDNoWaitInfo	1013
voiceOTASPStatusInfo	1013
voicePrivacyInfo	1014
voiceSetAllCallStatusCbkInfo	1015
voiceSetCallBarringPwdInfo	1017
voiceSetCallBarringPwdResp	1018
voiceSetConfigReq	1019
voiceSetConfigResp	1021
voiceSetPrefPrivacy	1022
voiceSetSUPSServiceReq	1023
voiceSetSUPSServiceResp	1025
voiceStopContDTMFInfo	1026
voiceSUPSInfo	1027
voiceSUPSNotification	1029
wcdmaCellInfo	1031
WCDMAECIOThresh	1032
WCDMAInfoLTENeighborCell	1032
wcdmaLongMsgDecodingParams	1033
wcdmaMsgDecodingParams	1035
wcdmaMsgEncodingParams	1036
WCDMARSSIThresh	1037
WCDMASysInfo	1038
wcdmaUARFCN	1041
wds_currNetworkInfo	1042
wds_Domain	1043
wds_DomainNameList	1044
wds_GPRSQoS	1044
wds_IPV6AddressInfo	1045
wds_IPV6GWAddressInfo	1045
wds_PCSCFFQDNAddress	1046
wds_PCSCFFQDNAddressList	1046
wds_PCSCFIPv4ServerAddressList	1047
wds_ProfileIdentifier	1047
wds_profileInfo	1048
wds_UMTSMinQoS	1048
WdsByteTotals	1051
WdsByteTotalsElmnts	1052
WdsClientLeaseChange	1052
WdsConnectionRate	1052
WdsConnectionRateElmnts	1053
WdsDHCPv4ClientLeaseInd	1054
WdsDHCPv4Config	1055
wdsDhcpv4HwConfig	1056
WdsDHCPv4HWConfig	1056
WdsDHCPv4Option	1057
wdsDhcpv4Option	1058
wdsDhcpv4OptionList	1058

WdsDHCPv4OptionList . . . . .	1059
WdsDHCPv4ProfileId . . . . .	1059
wdsDhcpv4ProfileId . . . . .	1060
WDSGetLoopbackData . . . . .	1060
WdsIpAddressInfoReq . . . . .	1061
WdsPktStatisticsElmnts . . . . .	1061
WdsPktStatisticsReq . . . . .	1063
WdsPktStatisticsResp . . . . .	1063
WdsProfileParam . . . . .	1064
WdsRunTimeSettings . . . . .	1064
wdsSetEventReportReq . . . . .	1065
WDSSetLoopbackData . . . . .	1067
WDSSWICurrentChannelRates . . . . .	1067

## 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 . . . . .	1069
<a href="#">common.h</a>		1069
<a href="#">dms.h</a>		1073
<a href="#">fms.h</a>		1107
<a href="#">loc.h</a>		1110
<a href="#">nas.h</a>		1121
<a href="#">qaCbkCatEventReportInd.h</a>		1149
<a href="#">qaCbkSwiOmaDmEventReportInd.h</a>		1150
<a href="#">qaGobiApiAudio.h</a>	Audio Service API function prototypes . . . . .	1151
<a href="#">qaGobiApiCat.h</a>	Card Application Toolkit API function headers . . . . .	1155
<a href="#">qaGobiApiCbk.h</a>	Callback Service API function prototypes . . . . .	1156
<a href="#">qaGobiApiDcs.h</a>	Device Connectivity Service API function prototypes . . . . .	1237
<a href="#">qaGobiApiDms.h</a>	Device Management Service API function prototypes . . . . .	1245
<a href="#">qaGobiApiFms.h</a>	Firmware Management Service API function prototypes . . . . .	1283
<a href="#">qaGobiApiIms.h</a>	IMS Service API function prototypes . . . . .	1300
<a href="#">qaGobiApiImsa.h</a>	IMSA Service API function prototypes . . . . .	1307
<a href="#">qaGobiApiLoc.h</a>	Location API function prototypes . . . . .	1310
<a href="#">qaGobiApiNas.h</a>	Network Access Service API function prototypes . . . . .	1316
<a href="#">qaGobiApiOmadm.h</a>	Open Mobile Alliance Device Management Service API function prototypes . . . . .	1363
<a href="#">qaGobiApiPds.h</a>	Position Determination Service API function prototypes . . . . .	1366
<a href="#">qaGobiApiQos.h</a>	Quality of Service API function prototypes . . . . .	1381
<a href="#">qaGobiApiRms.h</a>	Remote Management Service API function prototypes . . . . .	1387

<a href="#">qaGobiApiSar.h</a>	
Specific Absorption Rate API function prototypes	1388
<a href="#">qaGobiApiSms.h</a>	
Short Message Service API function prototypes	1391
<a href="#">qaGobiApiSwi.h</a>	
SWI API function prototypes	1413
<a href="#">qaGobiApiSwiAudio.h</a>	
M2M Audio Service API function prototypes	1414
<a href="#">qaGobiApiSwiOmadms.h</a>	
SWI Open Mobile Alliance Device Management Service API function prototypes	
QMI Service revision 1.6	1420
<a href="#">qaGobiApiTableBandClasses.h</a>	
Network Access Service API Band Classes table	1430
<a href="#">qaGobiApiTableCallControlReturnReasons.h</a>	
Call Control Return Reasons table	1433
<a href="#">qaGobiApiTableCallEndReasons.h</a>	
Wireless Data Service Call End Reasons	1434
<a href="#">qaGobiApiTableCarrierCodes.h</a>	
Carrier Codes table	1449
<a href="#">qaGobiApiTableCodingScheme.h</a>	
Data Coding Scheme	1451
<a href="#">qaGobiApiTableGpsCapabilityCodes.h</a>	
Position Determination Service API GPS Capability Codes	1454
<a href="#">qaGobiApiTablePowerModes.h</a>	
Device Management Service API Power Modes table	1454
<a href="#">qaGobiApiTableRadioInterfaces.h</a>	
Network Access Service API Radio Interfaces table	1455
<a href="#">qaGobiApiTableRegionCodes.h</a>	
Region Codes table	1456
<a href="#">qaGobiApiTableServiceOptions.h</a>	
Voice Service Options	1456
<a href="#">qaGobiApiTableSupServiceInfoClasses.h</a>	
Voice Supplementary Service Information Classes	1458
<a href="#">qaGobiApiTableSwiAudio.h</a>	
Swi Audio related tables	1459
<a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a>	
Update Complete Status table	1459
<a href="#">qaGobiApiTableVoiceCallEndReasons.h</a>	
Voice Service Call and supplementary services end reasons	1461
<a href="#">qaGobiApiTmd.h</a>	
Thermal Mitigation Device API function prototypes	1467
<a href="#">qaGobiApiUim.h</a>	
Uim Service API function prototypes	1469
<a href="#">qaGobiApiVoice.h</a>	
Voice Service API function prototypes	1482
<a href="#">qaGobiApiWds.h</a>	
Wireless Data Service API function prototypes	1502
<a href="#">qaNasGetRFBandInfo.h</a>	1544
<a href="#">qaNasPerformNetworkScan.h</a>	1545
<a href="#">qmerrno.h</a>	1546
<a href="#">qos.h</a>	1553
<a href="#">sms.h</a>	1561
<a href="#">SwiDataTypes.h</a>	
SWI data types	1570
<a href="#">swiloc.h</a>	1571
<a href="#">swioma.h</a>	1573
<a href="#">SWIWWANCMAPI.h</a>	1582
<a href="#">uim.h</a>	1582

<a href="#">wds.h</a>	1591
-----------------------	------



## 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

## 6.22 Thermal Mitigation Device(TMD)

### Files

- file [qaGobiApiTmd.h](#)  
*Thermal Mitigation Device API function prototypes.*

### 6.22.1 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>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>

## 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 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 \_getResetInfoNotification Struct Reference

### Data Fields

- [BYTE type](#)
- [BYTE source](#)

### 8.4.1 Detailed Description

Contains the parameters passed for SLQSSetSwtGetResetInfoCallback by the device.

#### Parameters

<i>type</i>	<ul style="list-style-type: none"> <li>• type of reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - warm</li> <li>– 2 - hard</li> <li>– 3 - crash</li> <li>– 4 - power down</li> </ul> </li> </ul>
<i>source</i>	<ul style="list-style-type: none"> <li>• entity which initiated the reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - user requested</li> <li>– 2 - hardware switch</li> <li>– 3 - temperature critical</li> <li>– 4 - voltage critical</li> <li>– 5 - configuration update</li> <li>– 6 - LWM2M</li> <li>– 7 - OMA-DM</li> <li>– 8 - FOTA</li> </ul> </li> </ul>

#### Note

None

## 8.4.2 Field Documentation

8.4.2.1 `BYTE _getResetInfoNotification::source`

8.4.2.2 `BYTE _getResetInfoNotification::type`

## 8.5 `_getTransLayerInfoResp` Struct Reference

### Data Fields

- `BYTE * pRegInd`
- `transLayerInfo * pTransLayerInfo`

### 8.5.1 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 <code>SLQSGetTransLayerInfo()</code> 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 <code>transLayerInfo</code>. <ul style="list-style-type: none"> <li>– Optional parameter</li> <li>– See <code>transLayerInfo</code> for more information</li> </ul> </li> <li>• function <code>SLQSGetTransLayerInfo()</code> returns a default value 0xFF for parameter values if no response is received from the device.</li> </ul>

## 8.5.2 Field Documentation

8.5.2.1 `BYTE* _getTransLayerInfoResp::pRegInd`

8.5.2.2 `transLayerInfo* _getTransLayerInfoResp::pTransLayerInfo`

## 8.6 `_getTransNWRegInfoResp` Struct Reference

### Data Fields

- `BYTE * pRegStatus`

### 8.6.1 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>
-------------------	---

## 8.6.2 Field Documentation

## 8.6.2.1 BYTE\* \_getTransNWRegInfoResp::pRegStatus

## 8.7 \_MitigationDevInfo Struct Reference

## Data Fields

- [BYTE deviceIdLen](#)
- [CHAR deviceId](#) [255]

## 8.7.1 Detailed Description

This structure contains mitigation Level Indication request parameters

## Parameters

<i>deviceIdLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements <ul style="list-style-type: none"> <li>deviceId</li> </ul> </li> </ul>
<i>deviceId</i>	<ul style="list-style-type: none"> <li>Mitigation device ID</li> </ul>

## 8.7.2 Field Documentation

## 8.7.2.1 CHAR \_MitigationDevInfo::deviceId[255]

## 8.7.2.2 BYTE \_MitigationDevInfo::deviceIdLen

## 8.8 \_modemTempNotification Struct Reference

## Data Fields

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

### 8.8.1 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

### 8.8.2 Field Documentation

8.8.2.1 **WORD \_modemTempNotification::ModemTemperature**

8.8.2.2 **BYTE \_modemTempNotification::ModemTempState**

## 8.9 \_packetSrvStatus Struct Reference

#### Data Fields

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

### 8.9.1 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.

## 8.9.2 Field Documentation

8.9.2.1 **BYTE** `_packetSrvStatus::bearerID`

8.9.2.2 **BYTE** `_packetSrvStatus::connStatus`

8.9.2.3 **BYTE** `_packetSrvStatus::ipFamily`

8.9.2.4 **qaQmiInterfaceInfo\*** `_packetSrvStatus::pQmiInterfaceInfo`

8.9.2.5 **BYTE** `_packetSrvStatus::reconfigReqd`

8.9.2.6 **WORD** `_packetSrvStatus::sessionEndReason`

8.9.2.7 **WORD** `_packetSrvStatus::techName`

8.9.2.8 **WORD** `_packetSrvStatus::verboseSessnEndReason`

8.9.2.9 **WORD** `_packetSrvStatus::verboseSessnEndReasonType`

## 8.10 `_qaQmi3GPP2BroadcastCfgInfo` Struct Reference

### Data Fields

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

### 8.10.1 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>

### 8.10.2 Field Documentation

8.10.2.1 **BYTE** \_qaQmi3GPP2BroadcastCfgInfo::activated\_ind

8.10.2.2 **struct** CDMABroadcastConfig \_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]

8.10.2.3 **WORD** \_qaQmi3GPP2BroadcastCfgInfo::num\_instances

## 8.11 \_qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- [BYTE](#) activated\_ind
- [WORD](#) num\_instances
- [struct BroadcastConfig](#) broadcastConfig [0x05]

### 8.11.1 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>

### 8.11.2 Field Documentation

8.11.2.1 **BYTE** \_qaQmi3GPPBroadcastCfgInfo::activated\_ind

8.11.2.2 **struct** BroadcastConfig \_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]

8.11.2.3 **WORD** \_qaQmi3GPPBroadcastCfgInfo::num\_instances

## 8.12 \_setIndicationRegReq Struct Reference

### Data Fields

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

- [BYTE \\* pRegCallStatInfoEvt](#)

### 8.12.1 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>

### 8.12.2 Field Documentation

8.12.2.1 [BYTE\\* \\_setIndicationRegReq::pRegCallStatInfoEvt](#)

8.12.2.2 [BYTE\\* \\_setIndicationRegReq::pRegTransLayerInfoEvt](#)

8.12.2.3 [BYTE\\* \\_setIndicationRegReq::pRegTransNWRegInfoEvt](#)

## 8.13 \_slqs3GPPConfigItem Struct Reference

#### Data Fields

- [WORD \\* pLTEAttachProfile](#)
- [WORD \\* pProfileList](#)
- [BYTE \\* pDefaultPDNEnabled](#)
- [BYTE \\* p3gppRelease](#)
- [WORD LTEAttachProfileListLen](#)
- [WORD \\* pLTEAttachProfileList](#)

### 8.13.1 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 (EM/MC73xx or earlier)</li> <li>– Optional parameter with possible values 1-24 (EM/MC74xx onwards)</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> <li>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfile-List". Please provide attach profiles in order of decreasing priority in this list.</li> </ul> </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>
--------------------------------	--

### 8.13.2 Field Documentation

8.13.2.1 **WORD** \_slqs3GPPConfigItem::LTEAttachProfileListLen

8.13.2.2 **BYTE\*** \_slqs3GPPConfigItem::p3gppRelease

8.13.2.3 **BYTE\*** \_slqs3GPPConfigItem::pDefaultPDNEnabled

8.13.2.4 **WORD\*** \_slqs3GPPConfigItem::pLTEAttachProfile

8.13.2.5 **WORD\*** \_slqs3GPPConfigItem::pLTEAttachProfileList

8.13.2.6 **WORD\*** \_slqs3GPPConfigItem::pProfileList

## 8.14 \_SlqsNas3GppNetworkRAT\_ Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [BYTE RAT](#)

### 8.14.1 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> <li>0x08 - LTE</li> <li>0x09 - TD-SCDMA</li> </ul> </li> </ul>

### 8.14.2 Field Documentation

8.14.2.1 **WORD** \_SlqsNas3GppNetworkRAT\_::MCC

8.14.2.2 WORD \_SlqsNas3GppNetworkRAT\_::MNC

8.14.2.3 BYTE \_SlqsNas3GppNetworkRAT\_::RAT

## 8.15 \_slqsNetworkScanInfo Struct Reference

### Data Fields

- [BYTE \\* pNetworkInfoInstances](#)
- [struct SlqsNas3GppNetworkInfo \\* pNetworkInfo](#)
- [BYTE \\* pRATInstances](#)
- [SlqsNas3GppNetworkRAT \\* pRATInfo](#)
- [BYTE \\* pPCSDigitInstances](#)
- [struct SlqsNasPcsDigit \\* pPCSDigitInfo](#)
- [ULONG \\* pScanResult](#)

### 8.15.1 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> [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>

### 8.15.2 Field Documentation

8.15.2.1 struct SlqsNas3GppNetworkInfo\* \_slqsNetworkScanInfo::pNetworkInfo

8.15.2.2 BYTE\* \_slqsNetworkScanInfo::pNetworkInfoInstances

8.15.2.3 struct SlqsNasPcsDigit\* \_slqsNetworkScanInfo::pPCSDigitInfo

8.15.2.4 BYTE\* \_slqsNetworkScanInfo::pPCSDigitInstances

8.15.2.5 SlqsNas3GppNetworkRAT\* \_slqsNetworkScanInfo::pRATInfo

8.15.2.6 BYTE\* \_slqsNetworkScanInfo::pRATInstances

8.15.2.7 ULONG\* \_slqsNetworkScanInfo::pScanResult

## 8.16 \_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
- WORD \* pRetryCount

### 8.16.1 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(optional)               <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(optional) <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(optional) <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.(optional)</li> </ul>
<i>pSource</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII(optional)</li> </ul>
<i>pPkgName-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.(optional)</li> </ul>
<i>pPkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII(optional)</li> </ul>
<i>pPkgDesc-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.(optional)</li> </ul>
<i>pPkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII(optional)</li> </ul>
<i>pDateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.(optional)</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.(optional)</li> </ul>
<i>pTime</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII(optional)</li> </ul>
<i>pSessionType</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter reflects the last session started for Sprint(optional) <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(optional) <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(optional) <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.

**8.16.2 Field Documentation**

8.16.2.1 **BYTE\*** \_SLQSOMADMSessionInfo::pDate

8.16.2.2 **WORD\*** \_SLQSOMADMSessionInfo::pDateLength

8.16.2.3 **WORD\*** \_SLQSOMADMSessionInfo::pPkgDescLength

8.16.2.4 **BYTE\*** \_SLQSOMADMSessionInfo::pPkgDescription

8.16.2.5 **BYTE\*** \_SLQSOMADMSessionInfo::pPkgName

8.16.2.6 **WORD\*** \_SLQSOMADMSessionInfo::pPkgNameLength

8.16.2.7 **WORD\*** \_SLQSOMADMSessionInfo::pRetryCount

8.16.2.8 **BYTE\*** \_SLQSOMADMSessionInfo::pSessionState

8.16.2.9 **BYTE\*** \_SLQSOMADMSessionInfo::pSessionType

8.16.2.10 **BYTE\*** \_SLQSOMADMSessionInfo::pSeverity

8.16.2.11 **BYTE\*** \_SLQSOMADMSessionInfo::pSource

8.16.2.12 **WORD\*** \_SLQSOMADMSessionInfo::pSourceLength

8.16.2.13 **BYTE\*** \_SLQSOMADMSessionInfo::pStatus

8.16.2.14 **BYTE\*** \_SLQSOMADMSessionInfo::pTime

8.16.2.15 WORD\* \_SLQSOMADMSessionInfo::pTimeLength

8.16.2.16 WORD\* \_SLQSOMADMSessionInfo::pUpdateCompleteStatus

## 8.17 \_SLQSOMADMSettings Struct Reference

### Data Fields

- ULONG \* pOMADMEEnabled
- BYTE \* pFOTAdownload
- BYTE \* pFOTAUpdate
- BYTE \* pAutosdm
- BYTE \* pFwAutoCheck

### 8.17.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

#### Parameters

<i>pOMADM-Enabled[OUT]</i>	<ul style="list-style-type: none"> <li>• 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>pFOTAdownload[OUT]</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 - 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>• 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>• 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

### 8.17.2 Field Documentation

8.17.2.1 **BYTE\*** \_SLQSOMADMSettings::pAutosdm

8.17.2.2 **BYTE\*** \_SLQSOMADMSettings::pFOTAdownload

8.17.2.3 **BYTE\*** \_SLQSOMADMSettings::pFOTAupdate

8.17.2.4 **BYTE\*** \_SLQSOMADMSettings::pFwAutoCheck

8.17.2.5 **ULONG\*** \_SLQSOMADMSettings::pOMADMEabled

## 8.18 \_SLQSOMADMSettingsReqParams Struct Reference

### Data Fields

- [BYTE FOTAdownload](#)
- [BYTE FOTAupdate](#)
- [BYTE \\*](#) pAutosdm

### 8.18.1 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

## 8.18.2 Field Documentation

8.18.2.1 **BYTE** \_SLQSOMADMSettingsReqParams::FOTAdownload

8.18.2.2 **BYTE** \_SLQSOMADMSettingsReqParams::FOTAUpdate

8.18.2.3 **BYTE\*** \_SLQSOMADMSettingsReqParams::pAutosdm

## 8.19 \_SLQSOMADMSettingsReqParams3 Struct Reference

## Data Fields

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

## 8.19.1 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

**8.19.2 Field Documentation**

8.19.2.1 **BYTE** \_SLQSOMADMSettingsReqParams3::FOTAdownload

8.19.2.2 **BYTE** \_SLQSOMADMSettingsReqParams3::FOTAUpdate

8.19.2.3 **BYTE\*** \_SLQSOMADMSettingsReqParams3::pAutosdm

8.19.2.4 **BYTE\*** \_SLQSOMADMSettingsReqParams3::pFwAutoCheck

**8.20 \_SLQSSwiGetHostDevInfoParams Struct Reference****Data Fields**

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

**8.20.1 Detailed Description**

This structure is used to Get Host Device Information

## Parameters

<i>bManSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Host Device Manufacturer String Size</li> </ul>
<i>pManString</i> [OUT]	<ul style="list-style-type: none"> <li>Host Device Manufacturer Name(Optional parameter)</li> <li>Null terminated ASCII String</li> </ul>
<i>bModelSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Host Device Model String Size</li> </ul>
<i>pModelString</i> [OUT]	<ul style="list-style-type: none"> <li>Host Device Model String(Optional parameter)</li> <li>Null terminated ASCII string.</li> </ul>
<i>bSWVerSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Host Device Software Version String Size</li> </ul>
<i>pSWVerString</i> [OUT]	<ul style="list-style-type: none"> <li>Host Device Software Version String(Optional parameter)</li> <li>Null terminated ASCII string</li> </ul>
<i>bPlasmaIDSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Host Device Plasma ID String Size</li> </ul>
<i>pPlasmaIDString</i> [OUT]	<ul style="list-style-type: none"> <li>Host Device Plasma ID String(Optional parameter)</li> <li>Null terminated alphanumeric ASCII String.</li> </ul>

## 8.20.2 Field Documentation

8.20.2.1 BYTE \_SLQSSwiGetHostDevInfoParams::bManSize

8.20.2.2 BYTE \_SLQSSwiGetHostDevInfoParams::bModelSize

8.20.2.3 BYTE \_SLQSSwiGetHostDevInfoParams::bPlasmaIDSize

8.20.2.4 BYTE \_SLQSSwiGetHostDevInfoParams::bSWVerSize

8.20.2.5 CHAR\* \_SLQSSwiGetHostDevInfoParams::pManString

8.20.2.6 CHAR\* \_SLQSSwiGetHostDevInfoParams::pModelString

8.20.2.7 CHAR\* \_SLQSSwiGetHostDevInfoParams::pPlasmaIDString

8.20.2.8 CHAR\* \_SLQSSwiGetHostDevInfoParams::pSWVerString

## 8.21 \_SLQSSwiGetOSInfoParams Struct Reference

## Data Fields

- [BYTE bNameSize](#)

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

### 8.21.1 Detailed Description

This structure is used to Get OS Information

#### Parameters

<i>bNameSize</i> [IN-OUT]	<ul style="list-style-type: none"> <li>• Size of Operating System Name</li> </ul>
<i>pNameString</i> [OUT]	<ul style="list-style-type: none"> <li>• Operating System Name(Optional parameter)</li> <li>• Null terminated ASCII string</li> </ul>
<i>bVersionSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Operating System Version Size</li> </ul>
<i>pVersionString</i> [OUT]	<ul style="list-style-type: none"> <li>• Operating System Version String(Optional parameter)</li> <li>• Null terminated ASCII string.</li> </ul>

### 8.21.2 Field Documentation

8.21.2.1 [BYTE \\_SLQSSwiGetOSInfoParams::bNameSize](#)

8.21.2.2 [BYTE \\_SLQSSwiGetOSInfoParams::bVersionSize](#)

8.21.2.3 [CHAR\\* \\_SLQSSwiGetOSInfoParams::pNameString](#)

8.21.2.4 [CHAR\\* \\_SLQSSwiGetOSInfoParams::pVersionString](#)

## 8.22 \_SLQSSwiGetSerialNoExtParams Struct Reference

#### Data Fields

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

### 8.22.1 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>• Optional parameter</li> <li>• Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device.</li> </ul>
--------------------------	---

## 8.22.2 Field Documentation

8.22.2.1 **BYTE** \_SLQSSwiGetSerialNoExtParams::meidLength

8.22.2.2 **CHAR\*** \_SLQSSwiGetSerialNoExtParams::pMeidString

## 8.23 \_SLQSSwiSetHostDevInfoParams Struct Reference

### Data Fields

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

### 8.23.1 Detailed Description

This structure is used to Set Host Device Information

#### Parameters

<i>bManSize</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Manufacturer String Size</li> </ul>
<i>pManString</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Manufacturer Name(Optional parameter)</li> <li>• Null terminated ASCII String</li> </ul>
<i>bModelSize</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Model String Size</li> </ul>
<i>pModelString</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Model String(Optional parameter)</li> <li>• Null terminated ASCII string.</li> </ul>
<i>bSWVerSize</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Software Version String Size</li> </ul>
<i>pSWVerString</i> [IN]	<ul style="list-style-type: none"> <li>• Host Device Software Version String(Optional parameter)</li> <li>• Null terminated ASCII string</li> </ul>

<i>bPlasmaIDSize</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Plasma ID String Size</li> </ul>
<i>pPlasmaIDString</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Plasma ID String(Optional parameter)</li> <li>Null terminated alphanumeric ASCII String.</li> </ul>

### 8.23.2 Field Documentation

8.23.2.1 **BYTE** \_SLQSSwiSetHostDevInfoParams::bManSize

8.23.2.2 **BYTE** \_SLQSSwiSetHostDevInfoParams::bModelSize

8.23.2.3 **BYTE** \_SLQSSwiSetHostDevInfoParams::bPlasmaIDSize

8.23.2.4 **BYTE** \_SLQSSwiSetHostDevInfoParams::bSWVerSize

8.23.2.5 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pManString

8.23.2.6 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pModelString

8.23.2.7 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pPlasmaIDString

8.23.2.8 **CHAR\*** \_SLQSSwiSetHostDevInfoParams::pSWVerString

## 8.24 \_SLQSSwiSetOSInfoParams Struct Reference

### Data Fields

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

### 8.24.1 Detailed Description

This structure is used to Set OS Information

#### Parameters

<i>bNameSize</i> [IN]	<ul style="list-style-type: none"> <li>Size of Operating System Name</li> </ul>
<i>pNameString</i> [IN]	<ul style="list-style-type: none"> <li>Operating System Name(Optional parameter)</li> <li>Null terminated ASCII string</li> </ul>
<i>bVersionSize</i> [IN]	<ul style="list-style-type: none"> <li>Operating System Version Size</li> </ul>

<i>pVersionString</i> [ <i>I-N</i> ]	<ul style="list-style-type: none"> <li>Operating System Version String(Optional parameter)</li> <li>Null terminated ASCII string.</li> </ul>
--------------------------------------	--

## 8.24.2 Field Documentation

8.24.2.1 **BYTE** \_SLQSSwiSetOSInfoParams::bNameSize

8.24.2.2 **BYTE** \_SLQSSwiSetOSInfoParams::bVersionSize

8.24.2.3 **CHAR\*** \_SLQSSwiSetOSInfoParams::pNameString

8.24.2.4 **CHAR\*** \_SLQSSwiSetOSInfoParams::pVersionString

## 8.25 \_sysSelectPrefInfo Struct Reference

### Data Fields

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

### 8.25.1 Detailed Description

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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.</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 22 - E-UTRA Operating Band 23</li> <li>– Bit 23 - E-UTRA Operating Band 24</li> <li>– Bit 24 - E-UTRA Operating Band 25</li> <li>– Bit 25 - E-UTRA Operating Band 26</li> <li>– Bit 27 - E-UTRA Operating Band 28</li> <li>– Bit 28 - E-UTRA Operating Band 29</li> <li>– Bit 29 - E-UTRA Operating Band 32</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> <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> <li>– Bit 40 - E-UTRA Operating Band 41</li> <li>– Bit 41 - E-UTRA Operating Band 42</li> <li>– Bit 42 - E-UTRA Operating Band 43</li> <li>– Bit 60 - E-UTRA Operating Band 125</li> <li>– All other bits are reserved</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.</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

### 8.25.2 Field Documentation

8.25.2.1 **ULONGLONG\*** \_sysSelectPrefInfo::pBandPref

8.25.2.2 **BYTE\*** \_sysSelectPrefInfo::pEmerMode

8.25.2.3 **ULONG\*** \_sysSelectPrefInfo::pGWAcqOrderPref

8.25.2.4 **ULONGLONG\*** \_sysSelectPrefInfo::pLTEBandPref

8.25.2.5 **WORD\*** \_sysSelectPrefInfo::pModePref

8.25.2.6 **BYTE\*** \_sysSelectPrefInfo::pNetSelPref

8.25.2.7 **WORD\*** \_sysSelectPrefInfo::pPRLPref

8.25.2.8 **WORD\*** \_sysSelectPrefInfo::pRoamPref

8.25.2.9 **ULONG\*** \_sysSelectPrefInfo::pSrvDomainPref

## 8.26 \_sysSelectPrefParams Struct Reference

### Data Fields

- [BYTE](#) \* [pEmerMode](#)
- [WORD](#) \* [pModePref](#)
- [ULONGLONG](#) \* [pBandPref](#)
- [WORD](#) \* [pPRLPref](#)
- [WORD](#) \* [pRoamPref](#)
- [ULONGLONG](#) \* [pLTETestBandPref](#)
- [struct](#) [netSelectionPref](#) \* [pNetSelPref](#)
- [BYTE](#) \* [pChgDuration](#)
- [BYTE](#) \* [pMNCIncPCSDigStat](#)
- [ULONG](#) \* [pSrvDomainPref](#)
- [ULONG](#) \* [pGWAacqOrderPref](#)
- [ULONGLONG](#) \* [pTdsdmaBandPref](#)
- [struct](#) [acqOrderPref](#) \* [pAcqOrderPref](#)
- [ULONG](#) \* [pSrvRegRestriction](#)
- [struct](#) [CSGID](#) \* [pCSGID](#)
- [BYTE](#) \* [pRAT](#)

### 8.26.1 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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</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 22 - E-UTRA Operating Band 23</li> <li>– Bit 23 - E-UTRA Operating Band 24</li> <li>– Bit 24 - E-UTRA Operating Band 25</li> <li>– Bit 25 - E-UTRA Operating Band 26</li> <li>– Bit 27 - E-UTRA Operating Band 28</li> <li>– Bit 28 - E-UTRA Operating Band 29</li> <li>– Bit 29 - E-UTRA Operating Band 32</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> <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> <li>– Bit 40 - E-UTRA Operating Band 41</li> <li>– Bit 41 - E-UTRA Operating Band 42</li> <li>– Bit 42 - E-UTRA Operating Band 43</li> <li>– Bit 60 - E-UTRA Operating Band 125</li> <li>– All other bits are reserved</li> </ul> </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>• Either of pNetSelPref or pCSGID can be set.</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>• At least one system selection setting to be set if pChgDuration is populated.</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>• pNetSelPref is expected if MNC includes PCS digit is set to 1.</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>pTdsdmaBand-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - TD-SCDMA Band A</li> <li>– 0x02 - TD-SCDMA Band B</li> <li>– 0x04 - TD-SCDMA Band C</li> <li>– 0x08 - TD-SCDMA Band D</li> <li>– 0x10 - TD-SCDMA Band E</li> <li>– 0x20 - TD-SCDMA Band F</li> <li>– All other bits are reserved</li> </ul> </li> </ul>

<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> <li>- <a href="#">acqOrderPref</a></li> <li>• Optional parameter for specifying Acquisition Order Preference</li> <li>• see <a href="#">acqOrderPref</a> for more information</li> </ul>
<i>pSrvReg-Restriction</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Network Selection Registration Restriction Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Device follows the normal registration process</li> <li>– 0x01 - Device camps on the network according to its provisioning, but does not register</li> <li>– 0x02 - Device selects the network for limited service</li> <li>– All other values are reserved.</li> </ul> </li> </ul>
<i>pCSGID</i>	<ul style="list-style-type: none"> <li>- <a href="#">CSGID</a></li> <li>• Optional parameter for specifying CSG ID</li> <li>• Either of pNetSelPref or pCSGID can be set.</li> <li>• see <a href="#">CSGID</a> for more information</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>

## 8.26.2 Field Documentation

8.26.2.1 `struct acqOrderPref* _sysSelectPrefParams::pAcqOrderPref`

8.26.2.2 `ULONGLONG* _sysSelectPrefParams::pBandPref`

8.26.2.3 `BYTE* _sysSelectPrefParams::pChgDuration`

8.26.2.4 `struct CSGID* _sysSelectPrefParams::pCSGID`

8.26.2.5 `BYTE* _sysSelectPrefParams::pEmerMode`

8.26.2.6 `ULONG* _sysSelectPrefParams::pGWAcqOrderPref`

8.26.2.7 `ULONGLONG* _sysSelectPrefParams::pLTEBandPref`

8.26.2.8 `BYTE* _sysSelectPrefParams::pMNCIncPCSDigStat`

8.26.2.9 `WORD* _sysSelectPrefParams::pModePref`

8.26.2.10 `struct netSelectionPref* _sysSelectPrefParams::pNetSelPref`

8.26.2.11 `WORD* _sysSelectPrefParams::pPRLPref`

8.26.2.12 `BYTE* _sysSelectPrefParams::pRAT`

8.26.2.13 WORD\* \_sysSelectPrefParams::pRoamPref

8.26.2.14 ULONG\* \_sysSelectPrefParams::pSrvDomainPref

8.26.2.15 ULONG\* \_sysSelectPrefParams::pSrvRegRestriction

8.26.2.16 ULONGLONG\* \_sysSelectPrefParams::pTdsdmaBandPref

## 8.27 \_transLayerinfo Struct Reference

### Data Fields

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

### 8.27.1 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>

### 8.27.2 Field Documentation

8.27.2.1 BYTE \_transLayerinfo::TransCap

8.27.2.2 BYTE \_transLayerinfo::TransType

## 8.28 \_transLayerInfoNotification Struct Reference

### Data Fields

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

### 8.28.1 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

## 8.28.2 Field Documentation

8.28.2.1 `transLayerInfo* _transLayerInfoNotification::pTransLayerInfo`8.28.2.2 `BYTE _transLayerInfoNotification::regInd`8.29 `_transNWRegInfoNotification` Struct Reference

## Data Fields

- [BYTE NWRegStat](#)

## 8.29.1 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

## 8.29.2 Field Documentation

8.29.2.1 `BYTE _transNWRegInfoNotification::NWRegStat`

## 8.30 accelAcceptReady\_s Struct Reference

### Data Fields

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

### 8.30.1 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>

### 8.30.2 Field Documentation

8.30.2.1 **WORD** accelAcceptReady\_s::batchPerSec

8.30.2.2 **BYTE** accelAcceptReady\_s::injectEnable

8.30.2.3 **WORD** accelAcceptReady\_s::samplesPerBatch

## 8.31 accelTempAcceptReady\_s Struct Reference

### Data Fields

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

### 8.31.1 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>

## 8.31.2 Field Documentation

8.31.2.1 WORD accelTempAcceptReady\_s::batchPerSec

8.31.2.2 BYTE accelTempAcceptReady\_s::injectEnable

8.31.2.3 WORD accelTempAcceptReady\_s::samplesPerBatch

## 8.32 acqOrderPref Struct Reference

## Data Fields

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

## 8.32.1 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>

### 8.32.2 Field Documentation

8.32.2.1 **BYTE** acqOrderPref::acqOrdeLen

8.32.2.2 **BYTE\*** acqOrderPref::pAcqOrder

## 8.33 ActPilotPNElement Struct Reference

### Data Fields

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

### 8.33.1 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>

### 8.33.2 Field Documentation

8.33.2.1 **WORD** ActPilotPNElement::ActSetPilotPN

8.33.2.2 **BYTE** ActPilotPNElement::ActSetPilotPNStrength

## 8.34 AddCDMASysInfo Struct Reference

### Data Fields

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

### 8.34.1 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>

### 8.34.2 Field Documentation

8.34.2.1 WORD AddCDMASysInfo::geoSysIdx

8.34.2.2 WORD AddCDMASysInfo::regPrd

## 8.35 AddSysInfo Struct Reference

### Data Fields

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

### 8.35.1 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>

### 8.35.2 Field Documentation

8.35.2.1 ULONG AddSysInfo::cellBroadcastCap

8.35.2.2 WORD AddSysInfo::geoSysIdx

## 8.36 airTimer Struct Reference

### Data Fields

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

### 8.36.1 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>

## 8.36.2 Field Documentation

8.36.2.1 ULONG airTimer::airTimerValue

8.36.2.2 BYTE airTimer::namID

## 8.37 allCallsAlphaIDInfo Struct Reference

## Data Fields

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

## 8.37.1 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>

## 8.37.2 Field Documentation

8.37.2.1 alphaIDInfo allCallsAlphaIDInfo::AlphaIDInfo

8.37.2.2 BYTE allCallsAlphaIDInfo::callID

## 8.38 allCallsDiagInfo Struct Reference

## Data Fields

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

### 8.38.1 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>

### 8.38.2 Field Documentation

8.38.2.1 **BYTE** `allCallsDiagInfo::callID`

8.38.2.2 **diagInfo** `allCallsDiagInfo::DiagInfo`

## 8.39 allCallsUUSInfo Struct Reference

#### Data Fields

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

### 8.39.1 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>

### 8.39.2 Field Documentation

8.39.2.1 **BYTE** `allCallsUUSInfo::callID`

8.39.2.2 **UUSInfo** `allCallsUUSInfo::uusInfo`

## 8.40 alphaIDInfo Struct Reference

#### Data Fields

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

### 8.40.1 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>

### 8.40.2 Field Documentation

8.40.2.1 **BYTE** alphaIDInfo::alphaDcs

8.40.2.2 **BYTE** alphaIDInfo::alphaLen

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

## 8.41 altitudeSrcInfo Struct Reference

### Data Fields

- [ULONG](#) source
- [ULONG](#) linkage
- [ULONG](#) coverage

### 8.41.1 Detailed Description

This structure specifies information regarding the altitude source

## Parameters

<i>source</i>	<ul style="list-style-type: none"> <li>• Specifies the source of the altitude</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Source is unknown</li> <li>– 1 - GPS is the source</li> <li>– 2 - Cell ID provided the source</li> <li>– 3 - Source is enhanced cell ID</li> <li>– 4 - Wi-Fi is the source</li> <li>– 5 - Terrestrial source</li> <li>– 6 - Hybrid terrestrial source</li> <li>– 7 - Altitude database is the source</li> <li>– 8 - Barometric altimeter is the source</li> <li>– 9 - Other sources</li> </ul> </li> </ul>
<i>linkage</i>	<ul style="list-style-type: none"> <li>• Specifies the dependency between the horizontal and altitude position components</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Not specified</li> <li>– 1 - Fully interdependent</li> <li>– 2 - Depends on latitude and longitude</li> <li>– 3 - Fully independent</li> </ul> </li> </ul>
<i>coverage</i>	<ul style="list-style-type: none"> <li>• Specifies the region of uncertainty.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Not specified</li> <li>– 1 - Altitude uncertainty is valid at the injected horizontal position coordinates only</li> <li>– 2 - Altitude uncertainty applies to the position of the device regardless of horizontal position</li> </ul> </li> </ul>

## 8.41.2 Field Documentation

8.41.2.1 **ULONG** altitudeSrcInfo::coverage8.41.2.2 **ULONG** altitudeSrcInfo::linkage8.41.2.3 **ULONG** altitudeSrcInfo::source

## 8.42 appStats Struct Reference

## Data Fields

- uint8\_t [appType](#)
- uint8\_t [appState](#)
- uint8\_t [persoState](#)
- uint8\_t [persoFeature](#)
- uint8\_t [persoRetries](#)
- uint8\_t [persoUnblockRetries](#)
- uint8\_t [aidLength](#)
- uint8\_t [aidVal](#) [255]
- uint8\_t [univPin](#)

- uint8\_t [pin1State](#)
- uint8\_t [pin1Retries](#)
- uint8\_t [puk1Retries](#)
- uint8\_t [pin2State](#)
- uint8\_t [pin2Retries](#)
- uint8\_t [puk2Retries](#)

### 8.42.1 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[MAX_DESCRIPTION_LENGTH]</i>	<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>

### 8.42.2 Field Documentation

8.42.2.1 `uint8_t appStats::aidLength`

8.42.2.2 `uint8_t appStats::aidVal[255]`

8.42.2.3 `uint8_t appStats::appState`

8.42.2.4 `uint8_t appStats::appType`

8.42.2.5 `uint8_t appStats::persoFeature`

8.42.2.6 `uint8_t appStats::persoRetries`

8.42.2.7 `uint8_t appStats::persoState`

8.42.2.8 `uint8_t appStats::persoUnblockRetries`

8.42.2.9 `uint8_t appStats::pin1Retries`

8.42.2.10 `uint8_t appStats::pin1State`

8.42.2.11 `uint8_t appStats::pin2Retries`

8.42.2.12 `uint8_t appStats::pin2State`

8.42.2.13 `uint8_t appStats::puk1Retries`

8.42.2.14 `uint8_t appStats::puk2Retries`

8.42.2.15 `uint8_t appStats::univPin`

## 8.43 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

### 8.43.1 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>

### 8.43.2 Field Documentation

8.43.2.1 **BYTE** appStatus::aidLength

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

8.43.2.3 **BYTE** appStatus::appState

8.43.2.4 **BYTE** appStatus::appType

8.43.2.5 **BYTE** appStatus::persoFeature

8.43.2.6 **BYTE** appStatus::persoRetries

8.43.2.7 **BYTE** appStatus::persoState

8.43.2.8 **BYTE** appStatus::persoUnblockRetries

8.43.2.9 **BYTE** appStatus::pin1Retries

8.43.2.10 **BYTE** appStatus::pin1State

8.43.2.11 **BYTE** appStatus::pin2Retries

8.43.2.12 **BYTE** appStatus::pin2State

8.43.2.13 **BYTE** appStatus::puk1Retries

8.43.2.14 **BYTE** appStatus::puk2Retries

8.43.2.15 **BYTE** appStatus::univPin

## 8.44 arrAlertingPattern Struct Reference

### Data Fields

- BYTE** numInstances

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

### 8.44.1 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</i> [MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>alertingPattern</i> [-MAX_NO_OF_CALLS]	<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>

### 8.44.2 Field Documentation

8.44.2.1 **ULONG** arrAlertingPattern::alertingPattern[20]

8.44.2.2 **BYTE** arrAlertingPattern::callID[20]

8.44.2.3 **BYTE** arrAlertingPattern::numInstances

## 8.45 arrAlertingType Struct Reference

#### Data Fields

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

### 8.45.1 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[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>AlertingType[MAX_NO_OF_CALLS]</i>	<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>

## 8.45.2 Field Documentation

8.45.2.1 **BYTE** arrAlertingType::AlertingType[20]8.45.2.2 **BYTE** arrAlertingType::callID[20]8.45.2.3 **BYTE** arrAlertingType::numInstances

## 8.46 arrAlphaID Struct Reference

## Data Fields

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

## 8.46.1 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>allCallsAlphaIDInfo[MAX_NO_OF_CALLS]</i>	<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>

## 8.46.2 Field Documentation

8.46.2.1 **allCallsAlphaIDInfo** arrAlphaID::allCallsAlphaIDInfoArr[20]8.46.2.2 **BYTE** arrAlphaID::numInstances

## 8.47 arrCalledPartyNum Struct Reference

## Data Fields

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

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

## 8.47.2 Field Documentation

8.47.2.1 [peerNumberInfo](#) arrCalledPartyNum::CalledPartyNum[20]

8.47.2.2 [BYTE](#) arrCalledPartyNum::numInstances

## 8.48 arrCallEndReason Struct Reference

## Data Fields

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

## 8.48.1 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[MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>callEndReason[MAX_NO_OF_CALLS]</i>	<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>

## 8.48.2 Field Documentation

8.48.2.1 **WORD** arrCallEndReason::callEndReason[20]

8.48.2.2 **BYTE** arrCallEndReason::callID[20]

8.48.2.3 **BYTE** arrCallEndReason::numInstances

## 8.49 arrCallInfo Struct Reference

### Data Fields

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

### 8.49.1 Detailed Description

This structure contains an array of Call Info

#### Parameters

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

## 8.49.2 Field Documentation

8.49.2.1 **getAllCallInformation** arrCallInfo::getAllCallInfo[20]

8.49.2.2 **BYTE** arrCallInfo::numInstances

## 8.50 arrConnectPartyNum Struct Reference

### Data Fields

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

### 8.50.1 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 <a href="#">ConnectedPartyNum</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
---------------------	--

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

## 8.50.2 Field Documentation

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

8.50.2.2 **BYTE** arrConnectPartyNum::numInstances

## 8.51 arrDiagInfo Struct Reference

### Data Fields

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

### 8.51.1 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>

## 8.51.2 Field Documentation

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

8.51.2.2 **BYTE** arrDiagInfo::numInstances

## 8.52 arrRedirPartyNum Struct Reference

### Data Fields

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

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

## 8.52.2 Field Documentation

8.52.2.1 `BYTE arrRedirPartyNum::numInstances`8.52.2.2 `peerNumberInfo arrRedirPartyNum::RedirPartyNum[20]`8.53 `arrRemotePartyName` Struct Reference

## Data Fields

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

## 8.53.1 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>

## 8.53.2 Field Documentation

8.53.2.1 `getAllCallRmtPtyName arrRemotePartyName::getAllCallRmtPtyName[20]`8.53.2.2 `BYTE arrRemotePartyName::numInstances`8.54 `arrRemotePartyNum` Struct Reference

## Data Fields

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

### 8.54.1 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[MA- X_NO_OF_CAL- LS]</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>

### 8.54.2 Field Documentation

8.54.2.1 **BYTE** arrRemotePartyNum::numInstances

8.54.2.2 **getAllCallRmtPtyNum** arrRemotePartyNum::RmtPtyNum[20]

## 8.55 arrSvcOption Struct Reference

#### Data Fields

- **BYTE** numInstances
- **BYTE** callID [20]
- **WORD** srvOption [20]

### 8.55.1 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[MAX_NO- _OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>srvOption[MAX- _NO_OF_CALL- S]</i>	<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>

### 8.55.2 Field Documentation

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

8.55.2.2 **BYTE** arrSvcOption::numInstances

8.55.2.3 WORD arrSvcOption::srvOption[20]

## 8.56 arrUUSInfo Struct Reference

### Data Fields

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

### 8.56.1 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>AllCallsUUS-Info[MAX_NO_OF_CALLS]</i>	<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>

### 8.56.2 Field Documentation

8.56.2.1 [allCallsUUSInfo arrUUSInfo::AllCallsUUSInfo](#)[20]

8.56.2.2 [BYTE arrUUSInfo::numInstances](#)

## 8.57 authenticateResult Struct Reference

### Data Fields

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

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

## 8.57.2 Field Documentation

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

8.57.2.2 **WORD** authenticateResult::contentLen

## 8.58 authenticationData Struct Reference

### Data Fields

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

### 8.58.1 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 aUSIM 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 aISIM application)</li> <li>– 12 - HTTP-digest security context (valid only onan ISIM application)</li> <li>– 13 - Compute IP authentication, CHAP (valid onlyon 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</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• Authenticate Data.</li> </ul>

## 8.58.2 Field Documentation

8.58.2.1 **BYTE** authenticationData::context

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

8.58.2.3 **WORD** authenticationData::dataLen

## 8.59 BandCapabilityResp Struct Reference

### Data Fields

- [ULONGLONG](#) bandCapability
- [ULONGLONG](#) \* pLteBandCapability
- [ULONGLONG](#) \* pTdsBandCapability

### 8.59.1 Detailed Description

This structure contains the TLV required to Get Band Capability.

### Parameters

---

<i>bandCapability[OUT]</i>	<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> <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> <li>• Bit 56 - Band class 16</li> <li>• Bit 57 - Band class 17</li> <li>• Bit 58 - Band class 18</li> <li>• Bit 59 - Band class 19</li> </ul>
----------------------------	---

<i>OUT]</i>	<p>pLteBandCapability[OUT] Bitmask of LTE bands supported by the device</p> <ul style="list-style-type: none"> <li>• Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170</li> <li>• Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990</li> <li>• Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880</li> <li>• Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155</li> <li>• Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894</li> <li>• Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885</li> <li>• Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690</li> <li>• Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960</li> <li>• Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9</li> <li>• Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170</li> <li>• Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9</li> <li>• Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746</li> <li>• Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756</li> <li>• Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768</li> <li>• Bits 14 and 15 - Reserved</li> <li>• Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746</li> <li>• Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875</li> <li>• Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890</li> <li>• Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821</li> <li>• Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9</li> <li>• Bit 21 - Reserved</li> <li>• Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200</li> <li>• Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559</li> <li>• Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995</li> <li>• Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894</li> <li>• Bit 26 - Reserved</li> <li>• Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803</li> <li>• Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728</li> <li>• Bits 29 through 31 - Reserved</li> <li>• Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920</li> <li>• Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025</li> <li>• Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910</li> <li>• Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990</li> <li>• Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930</li> <li>• Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620</li> <li>• Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920</li> <li>• Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400</li> <li>• Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690</li> <li>• Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600</li> <li>• Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800</li> <li>• Bits 43 through 64 - Reserved</li> </ul>
-------------	--

<i>OUT]</i>	<p>pTdsBandCapability[OUT] Bitmask of TDS bands supported by the device.</p> <ul style="list-style-type: none"> <li>• Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz</li> <li>• Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz</li> <li>• Bit 2 - TDS Band C 1910 to 1930 MHz</li> <li>• Bit 3 - TDS Band D 2570 to 2620 MHz</li> <li>• Bit 4 - TDS Band E 2300 to 2400 MHz</li> <li>• Bit 5 - TDS Band F 1880 to 1920 MHz</li> </ul>
-------------	---

## 8.59.2 Field Documentation

8.59.2.1 **ULONGLONG** BandCapabilityResp::bandCapability

8.59.2.2 **ULONGLONG\*** BandCapabilityResp::pLteBandCapability

8.59.2.3 **ULONGLONG\*** BandCapabilityResp::pTdsBandCapability

## 8.60 BdsSV Struct Reference

### Data Fields

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

### 8.60.1 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>

## 8.60.2 Field Documentation

8.60.2.1 **WORD** BdsSV::id

8.60.2.2 **BYTE** BdsSV::mask

## 8.61 BdsSVInfo Struct Reference

## Data Fields

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

### 8.61.1 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>

### 8.61.2 Field Documentation

#### 8.61.2.1 BYTE BdsSVInfo::len

#### 8.61.2.2 BdsSV\* BdsSVInfo::pSV

## 8.62 BroadcastConfig Struct Reference

## Data Fields

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

### 8.62.1 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>

### 8.62.2 Field Documentation

8.62.2.1 WORD BroadcastConfig::fromServiceId

8.62.2.2 BYTE BroadcastConfig::selected

8.62.2.3 WORD BroadcastConfig::toServiceId

## 8.63 burstDTMFInfo Struct Reference

### Data Fields

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

### 8.63.1 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>

### 8.63.2 Field Documentation

8.63.2.1 BYTE burstDTMFInfo::digitCnt

8.63.2.2 BYTE\* burstDTMFInfo::pCallID

8.63.2.3 BYTE burstDTMFInfo::pDigitBuff[255]

## 8.64 CallBarringSysInfo Struct Reference

### Data Fields

- ULONG csBarStatus
- ULONG psBarStatus

### 8.64.1 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>

### 8.64.2 Field Documentation

8.64.2.1 **ULONG** CallBarringSysInfo::csBarStatus

8.64.2.2 **ULONG** CallBarringSysInfo::psBarStatus

## 8.65 callBarStatus Struct Reference

### Data Fields

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

### 8.65.1 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>

## 8.65.2 Field Documentation

8.65.2.1 ULONG callBarStatus::csBarStatus

8.65.2.2 ULONG callBarStatus::psBarStatus

## 8.66 calledPartyInfo Struct Reference

## Data Fields

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

## 8.66.1 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>

## 8.66.2 Field Documentation

### 8.66.2.1 BYTE calledPartyInfo::number[255]

### 8.66.2.2 BYTE calledPartyInfo::numLen

### 8.66.2.3 BYTE calledPartyInfo::numPlan

8.66.2.4 BYTE calledPartyInfo::numType

8.66.2.5 BYTE calledPartyInfo::PI

8.66.2.6 BYTE calledPartyInfo::SI

## 8.67 calledPartySubAdd Struct Reference

### Data Fields

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

### 8.67.1 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</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• Array of the SubAddress in BCD number format.</li> </ul>

### 8.67.2 Field Documentation

8.67.2.1 BYTE calledPartySubAdd::extBit

8.67.2.2 BYTE calledPartySubAdd::oddEvenInd

8.67.2.3 BYTE calledPartySubAdd::subAddr[255]

8.67.2.4 BYTE calledPartySubAdd::subAddrLen

8.67.2.5 **BYTE** calledPartySubAdd::subAddrType

## 8.68 callerIDInfo Struct Reference

### Data Fields

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

### 8.68.1 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>

### 8.68.2 Field Documentation

8.68.2.1 **BYTE** callerIDInfo::callerID[255]

8.68.2.2 **BYTE** callerIDInfo::callerIDLen

8.68.2.3 **BYTE** callerIDInfo::PI

## 8.69 callFwdTypeAndPlan Struct Reference

### Data Fields

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

### 8.69.1 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>

## 8.69.2 Field Documentation

8.69.2.1 BYTE callFwdTypeAndPlan::numberPlan

8.69.2.2 BYTE callFwdTypeAndPlan::numberType

## 8.70 callFWExtInfo Struct Reference

## Data Fields

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

## 8.70.1 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>

## 8.70.2 Field Documentation

8.70.2.1 **BYTE** callFWExtInfo::noReplyTimer

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

8.70.2.3 **BYTE** callFWExtInfo::numLen

8.70.2.4 **BYTE** callFWExtInfo::numPlan

8.70.2.5 **BYTE** callFWExtInfo::numType

8.70.2.6 **BYTE** callFWExtInfo::PI

8.70.2.7 **BYTE** callFWExtInfo::SI

8.70.2.8 **BYTE** callFWExtInfo::SvcClass

8.70.2.9 **BYTE** callFWExtInfo::SvcStatus

## 8.71 callFWInfo Struct Reference

### Data Fields

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

### 8.71.1 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>

## 8.71.2 Field Documentation

8.71.2.1 **BYTE** callFWInfo::noReplyTimer

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

8.71.2.3 **BYTE** callFWInfo::numLen

8.71.2.4 **BYTE** callFWInfo::SvcClass

8.71.2.5 **BYTE** callFWInfo::SvcStatus

## 8.72 callInfo Struct Reference

### Data Fields

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

### 8.72.1 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>

## 8.72.2 Field Documentation

## 8.72.2.1 BYTE callInfo::callID

8.72.2.2 **BYTE** callInfo::callState

8.72.2.3 **BYTE** callInfo::callType

8.72.2.4 **BYTE** callInfo::direction

8.72.2.5 **BYTE** callInfo::mode

## 8.73 callingPartyInfo Struct Reference

### Data Fields

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

### 8.73.1 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>

## 8.73.2 Field Documentation

8.73.2.1 BYTE callingPartyInfo::number[255]

8.73.2.2 BYTE callingPartyInfo::numLen

8.73.2.3 BYTE callingPartyInfo::numPlan

8.73.2.4 BYTE callingPartyInfo::numType

8.73.2.5 BYTE callingPartyInfo::PI

8.73.2.6 BYTE callingPartyInfo::SI

## 8.74 cardResult Struct Reference

### Data Fields

- [BYTE sw1](#)

- [BYTE sw2](#)

### 8.74.1 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>

### 8.74.2 Field Documentation

#### 8.74.2.1 BYTE cardResult::sw1

#### 8.74.2.2 BYTE cardResult::sw2

## 8.75 cardStatus Struct Reference

### Data Fields

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

### 8.75.1 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[<a href="#">MAX_N-O_OF_SLOTS</a>]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slotInfo</a> for more information.</li> </ul>

## 8.75.2 Field Documentation

8.75.2.1 WORD cardStatus::index1xPri

8.75.2.2 WORD cardStatus::index1xSec

8.75.2.3 WORD cardStatus::indexGwPri

8.75.2.4 WORD cardStatus::indexGwSec

8.75.2.5 BYTE cardStatus::numSlot

8.75.2.6 slotInfo cardStatus::SlotInfo[5]

## 8.76 CarrierImage\_t Struct Reference

### Data Fields

- uint32\_t [m\\_nCarrierId](#)
- uint32\_t [m\\_nFolderId](#)
- uint32\_t [m\\_nStorage](#)
- uint8\_t [m\\_FwImageId](#) [16]
- uint8\_t [m\\_FwBuildId](#) [32]
- uint8\_t [m\\_PriImageId](#) [16]
- uint8\_t [m\\_PriBuildId](#) [32]

### 8.76.1 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>

## 8.76.2 Field Documentation

8.76.2.1 `uint8_t CarrierImage_t::m_FwBuildId[32]`8.76.2.2 `uint8_t CarrierImage_t::m_FwImageId[16]`8.76.2.3 `uint32_t CarrierImage_t::m_nCarrierId`8.76.2.4 `uint32_t CarrierImage_t::m_nFolderId`8.76.2.5 `uint32_t CarrierImage_t::m_nStorage`8.76.2.6 `uint8_t CarrierImage_t::m_PriBuildId[32]`8.76.2.7 `uint8_t CarrierImage_t::m_PriImageId[16]`

## 8.77 CatAlPhalIdentifierTlv Struct Reference

## Data Fields

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

### 8.77.1 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]

### 8.77.2 Field Documentation

8.77.2.1 **BYTE** CatAlPhalIdentifierTlv::AlphaID[255]

8.77.2.2 **USHORT** CatAlPhalIdentifierTlv::AlphaDLength

8.77.2.3 **BYTE** CatAlPhalIdentifierTlv::ReferenceID

## 8.78 CatCommonEventTlv Struct Reference

### Data Fields

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

### 8.78.1 Field Documentation

8.78.1.1 union [currentCatEvent](#) CatCommonEventTlv::CatEvent

8.78.1.2 **BYTE** CatCommonEventTlv::EventID

8.78.1.3 **WORD** CatCommonEventTlv::EventLength

8.78.1.4 **BYTE** CatCommonEventTlv::TlvPresent

## 8.79 CatEndProactiveSessionTlv Struct Reference

### Data Fields

- [BYTE EndProactiveSession](#)

### 8.79.1 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>
-----------------------------	--

## 8.79.2 Field Documentation

8.79.2.1 **BYTE** CatEndProactiveSessionTlv::EndProactiveSession

## 8.80 CATEventDataType Struct Reference

### Data Fields

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

## 8.80.1 Field Documentation

8.80.1.1 **ULONG** CATEventDataType::eventMask

8.80.1.2 **ULONG\*** CATEventDataType::pErrorMask

## 8.81 CatEventIDDataTlv Struct Reference

### Data Fields

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

### 8.81.1 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]

## 8.81.2 Field Documentation

8.81.2.1 **BYTE** CatEventIDDataTlv::Data[255]

8.81.2.2 **USHORT** CatEventIDDataTlv::DataLength

8.81.2.3 **ULONG** CatEventIDDataTlv::ReferenceID

## 8.82 CatEventListTlv Struct Reference

### Data Fields

- [ULONG](#) SetupEventList

### 8.82.1 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>
-----------------------	---

### 8.82.2 Field Documentation

8.82.2.1 **ULONG** CatEventListTlv::SetupEventList

## 8.83 CatRefreshTlv Struct Reference

#### Data Fields

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

### 8.83.1 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>

### 8.83.2 Field Documentation

8.83.2.1 **USHORT** CatRefreshTlv::RefreshMode

8.83.2.2 **BYTE** CatRefreshTlv::RefreshStage

## 8.84 ccSUPSType Struct Reference

#### Data Fields

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

### 8.84.1 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>

### 8.84.2 Field Documentation

#### 8.84.2.1 BYTE ccSUPSType::reason

#### 8.84.2.2 BYTE ccSUPSType::svcType

## 8.85 CDMABroadcastConfig Struct Reference

### Data Fields

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

### 8.85.1 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>
-----------------	---

## 8.85.2 Field Documentation

8.85.2.1 WORD CDMABroadcastConfig::language

8.85.2.2 BYTE CDMABroadcastConfig::selected

8.85.2.3 WORD CDMABroadcastConfig::serviceCategory

## 8.86 CDMAChannel Struct Reference

### Data Fields

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

### 8.86.1 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>

## 8.86.2 Field Documentation

8.86.2.1 WORD CDMAChannel::priChA

8.86.2.2 WORD CDMAChannel::priChB

8.86.2.3 WORD CDMAChannel::secChA

8.86.2.4 WORD CDMAChannel::secChB

## 8.87 CDMAECIOThresh Struct Reference

### Data Fields

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

### 8.87.1 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>

### 8.87.2 Field Documentation

8.87.2.1 BYTE CDMAECIOThresh::CDMAECIOThreshListLen

8.87.2.2 WORD\* CDMAECIOThresh::pCDMAECIOThreshList

## 8.88 CDMAInfo Struct Reference

### Data Fields

- [WORD sid](#)
- [WORD nid](#)
- [WORD baseId](#)
- [WORD refpn](#)
- [ULONG baseLat](#)
- [ULONG baseLong](#)

### 8.88.1 Detailed Description

This structure contains information about the CDMA Network.

#### Parameters

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

## 8.88.2 Field Documentation

8.88.2.1 WORD CDMAInfo::baseId

8.88.2.2 ULONG CDMAInfo::baseLat

8.88.2.3 ULONG CDMAInfo::baseLong

8.88.2.4 WORD CDMAInfo::nid

8.88.2.5 WORD CDMAInfo::refpn

8.88.2.6 WORD CDMAInfo::sid

## 8.89 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`

### 8.89.1 Detailed Description

Structure contains parameters which need to be decoded from message

#### Parameters

<i>message- Length[IN]</i>	<ul style="list-style-type: none"> <li>• Length of the message to be decoded in bytes</li> </ul>
<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• Message read off the device via GetSMS</li> </ul>
<i>pSenderAddr- Length[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. 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[O- UT]</i>	<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>pTextMsg- Length[IN/OUT]</i>	<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[OUT]</i>	<ul style="list-style-type: none"> <li>• Returns the text message as NULL-terminated UCS2 string</li> </ul>
<i>pPriority[OUT]</i>	(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[OU- T](optional</i>	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[OU- T]</i>	(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>mcTime- Stamp[8][OUT]</i>	(optional parameter) <ul style="list-style-type: none"> <li>• Returns the message center timestamp which takes the form: YYMMDDHHMMSSTZ 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>absolute-Validity</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>pRelative-Validity</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</i> [OUT]	(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</i> [OUT]	(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</i> [OUT]	(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</i> [OUT]	(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</i> [OUT]	(optional parameter) <ul style="list-style-type: none"> <li>returns the length of Callback address string (including the NULL termination)</li> </ul>
<i>pCallbkAddr</i> [OUT]	(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>

## 8.89.2 Field Documentation

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

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

8.89.2.3 **ULONG** cdmaMsgDecodingParams::messageLength

8.89.2.4 **BYTE\*** cdmaMsgDecodingParams::pAlertPriority

8.89.2.5 **CHAR\*** cdmaMsgDecodingParams::pCallbkAddr

8.89.2.6 **BYTE\*** cdmaMsgDecodingParams::pCallbkAddrLength

8.89.2.7 **BYTE\*** cdmaMsgDecodingParams::pDisplayMode

- 8.89.2.8 **BYTE\*** cdmaMsgDecodingParams::pLanguage
- 8.89.2.9 **BYTE\*** cdmaMsgDecodingParams::pMessage
- 8.89.2.10 **ULONG\*** cdmaMsgDecodingParams::pMessageId
- 8.89.2.11 **BYTE\*** cdmaMsgDecodingParams::pPriority
- 8.89.2.12 **BYTE\*** cdmaMsgDecodingParams::pPrivacy
- 8.89.2.13 **BOOL\*** cdmaMsgDecodingParams::pReadAcknowledgementReq
- 8.89.2.14 **BYTE\*** cdmaMsgDecodingParams::pRelativeValidity
- 8.89.2.15 **CHAR\*** cdmaMsgDecodingParams::pSenderAddr
- 8.89.2.16 **BYTE\*** cdmaMsgDecodingParams::pSenderAddrLength
- 8.89.2.17 **WORD\*** cdmaMsgDecodingParams::pTextMsg
- 8.89.2.18 **BYTE\*** cdmaMsgDecodingParams::pTextMsgLength
- 8.89.2.19 **BOOL\*** cdmaMsgDecodingParams::pUserAcknowledgementReq

## 8.90 cdmaMsgEncodingParams Struct Reference

### Data Fields

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

### 8.90.1 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>messageld</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)	parameter) <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)	parameter) <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)	parameter) <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.

## 8.90.2 Field Documentation

8.90.2.1 **BYTE** cdmaMsgEncodingParams::messageld

8.90.2.2 **CHAR\*** cdmaMsgEncodingParams::pCallbackAddr

8.90.2.3 **CHAR\*** cdmaMsgEncodingParams::pDestAddr

8.90.2.4 **BYTE\*** cdmaMsgEncodingParams::pEncodingAlphabet

8.90.2.5 **BYTE\*** cdmaMsgEncodingParams::pMessage

8.90.2.6 **BYTE\*** cdmaMsgEncodingParams::pMessageSize

8.90.2.7 **BYTE\*** cdmaMsgEncodingParams::pPriority

8.90.2.8 **BYTE\*** cdmaMsgEncodingParams::pRelValidity

8.90.2.9 **WORD\*** cdmaMsgEncodingParams::pTextMsg

8.90.2.10 **ULONG** cdmaMsgEncodingParams::textMsgLength

## 8.91 CDMARSSIThresh Struct Reference

### Data Fields

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

### 8.91.1 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>

### 8.91.2 Field Documentation

8.91.2.1 **BYTE** CDMARSSIThresh::CDMARSSIThreshListLen

8.91.2.2 **WORD\*** CDMARSSIThresh::pCDMARSSIThreshList

## 8.92 CDMASSThresh Struct Reference

### Data Fields

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

### 8.92.1 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>
-------------	--

## 8.92.2 Field Documentation

### 8.92.2.1 SHORT CDMA SysInfo::ecio

### 8.92.2.2 INT8 CDMA SysInfo::rssi

## 8.93 cdma SSInfo Struct Reference

### Data Fields

- int8\_t [rssi](#)
- int16\_t [ecio](#)

### 8.93.1 Detailed Description

#### Parameters

<i>rssi</i>	RSSI in dBm.
<i>ecio</i>	ECIO value representing negative 0.5 dBm increment

## 8.93.2 Field Documentation

### 8.93.2.1 int16\_t cdma SSInfo::ecio

### 8.93.2.2 int8\_t cdma SSInfo::rssi

## 8.94 CDMA SysInfo Struct Reference

### Data Fields

- [sysInfoCommon sysInfoCDMA](#)
- [BYTE isSysPrlMatchValid](#)
- [BYTE isSysPrlMatch](#)
- [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]

### 8.94.1 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>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>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>ccsSupportedValid</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> <li>• MCC wildcard value is returned as {'3', 0xFF, 0xFF}.</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> <li>• MNC wildcard value is returned as {'7', 0xFF, 0xFF}.</li> </ul>

## 8.94.2 Field Documentation

8.94.2.1 WORD CDMA SysInfo::baseId

8.94.2.2 ULONG CDMA SysInfo::baseLat

8.94.2.3 ULONG CDMA SysInfo::baseLong

8.94.2.4 BYTE CDMA SysInfo::bsInfoValid

8.94.2.5 BYTE CDMA SysInfo::bsPRev

8.94.2.6 BYTE CDMA SysInfo::bsPRevValid

8.94.2.7 BYTE CDMA SysInfo::ccsSupported

8.94.2.8 BYTE CDMA SysInfo::ccsSupportedValid

8.94.2.9 BYTE CDMA SysInfo::cdmaSysIdValid

8.94.2.10 BYTE CDMA SysInfo::isSysPrIMatch

8.94.2.11 BYTE CDMA SysInfo::isSysPrIMatchValid

8.94.2.12 BYTE CDMA SysInfo::MCC[3]

8.94.2.13 BYTE CDMA SysInfo::MNC[3]

8.94.2.14 WORD CDMA SysInfo::networkId

8.94.2.15 **BYTE** CDMA SysInfo::networkIdValid

8.94.2.16 **WORD** CDMA SysInfo::packetZone

8.94.2.17 **BYTE** CDMA SysInfo::packetZoneValid

8.94.2.18 **BYTE** CDMA SysInfo::pRevInUse

8.94.2.19 **BYTE** CDMA SysInfo::pRevInUseValid

8.94.2.20 **sysInfoCommon** CDMA SysInfo::sysInfoCDMA

8.94.2.21 **WORD** CDMA SysInfo::systemID

## 8.95 CDMA SysInfoExt Struct Reference

### Data Fields

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

### 8.95.1 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>

### 8.95.2 Field Documentation

8.95.2.1 **BYTE** CDMA SysInfoExt::imsi\_11\_12

8.95.2.2 **WORD** CDMA SysInfoExt::MCC

## 8.96 CellDb Struct Reference

### Data Fields

- [ULONG mask](#)

### 8.96.1 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>
-------------	--

## 8.96.2 Field Documentation

## 8.96.2.1 ULONG CellDb::mask

## 8.97 cellParams Struct Reference

## Data Fields

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

## 8.97.1 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>
---------------	---

## 8.97.2 Field Documentation

8.97.2.1 **WORD** cellParams::pci

8.97.2.2 **SHORT** cellParams::rsrp

8.97.2.3 **SHORT** cellParams::rsrq

8.97.2.4 **SHORT** cellParams::rssi

8.97.2.5 **SHORT** cellParams::srxlev

## 8.98 changeUIMPIN Struct Reference

### Data Fields

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

### 8.98.1 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</i> [MAX- _DESCRIPTIO- N_LENGTH]	<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</i> [ <i>MAX_DESCRIPTION_LENGTH</i> ]	<ul style="list-style-type: none"> <li>• New PIN value.</li> <li>• This value is a sequence of ASCII characters.</li> </ul>
---	---

## 8.98.2 Field Documentation

8.98.2.1 **BYTE** *changeUIMPIN::oldPINLen*

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

8.98.2.3 **BYTE** *changeUIMPIN::pinID*

8.98.2.4 **BYTE** *changeUIMPIN::pinLen*

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

## 8.99 ChannelRate Struct Reference

### Data Fields

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

### 8.99.1 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>

## 8.99.2 Field Documentation

8.99.2.1 **ULONG** *ChannelRate::CurrChanRxRate*

8.99.2.2 **ULONG** *ChannelRate::CurrChanTxRate*

8.99.2.3 **ULONG** *ChannelRate::MaxChanRxRate*

8.99.2.4 **ULONG** ChannelRate::MaxChanTxRate

## 8.100 channelRate Struct Reference

### Data Fields

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

#### 8.100.1 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>

#### 8.100.2 Field Documentation

8.100.2.1 **ULONG** channelRate::CurrChanRxRate

8.100.2.2 **ULONG** channelRate::CurrChanTxRate

## 8.101 CLIPResp Struct Reference

### Data Fields

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

#### 8.101.1 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>
------------------------	--

## 8.101.2 Field Documentation

### 8.101.2.1 BYTE CLIPResp::ActiveStatus

### 8.101.2.2 BYTE CLIPResp::ProvisionStatus

## 8.102 CLIRResp Struct Reference

### Data Fields

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

### 8.102.1 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>

## 8.102.2 Field Documentation

### 8.102.2.1 BYTE CLIRResp::ActiveStatus

### 8.102.2.2 BYTE CLIRResp::ProvisionStatus

## 8.103 CkInfo Struct Reference

### Data Fields

- [ULONG mask](#)

### 8.103.1 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>
-------------	--

### 8.103.2 Field Documentation

#### 8.103.2.1 ULONG ClkInfo::mask

## 8.104 CNAPResp Struct Reference

### Data Fields

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

### 8.104.1 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>

### 8.104.2 Field Documentation

#### 8.104.2.1 BYTE CNAPResp::ActiveStatus

#### 8.104.2.2 BYTE CNAPResp::ProvisionStatus

## 8.105 COLPResp Struct Reference

### Data Fields

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

### 8.105.1 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>

## 8.105.2 Field Documentation

## 8.105.2.1 BYTE COLPResp::ActiveStatus

## 8.105.2.2 BYTE COLPResp::ProvisionStatus

## 8.106 COLRResp Struct Reference

## Data Fields

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

## 8.106.1 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>

## 8.106.2 Field Documentation

### 8.106.2.1 BYTE COLRResp::ActiveStatus

### 8.106.2.2 BYTE COLRResp::ProvisionStatus

## 8.107 CommInfo Struct Reference

### Data Fields

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

### 8.107.1 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>

## 8.107.2 Field Documentation

8.107.2.1 **BYTE** CommInfo::imsRegState

8.107.2.2 **BYTE** CommInfo::modemMode

8.107.2.3 **BYTE** CommInfo::psState

8.107.2.4 **BYTE** CommInfo::systemMode

8.107.2.5 **BYTE** CommInfo::temperature

## 8.108 ConnectionStatus Struct Reference

### Data Fields

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

### 8.108.1 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>

### 8.108.2 Field Documentation

8.108.2.1 **ULONGLONG** ConnectionStatus::MDMCallDuration

8.108.2.2 **BYTE** ConnectionStatus::MDMConnStatus

## 8.109 connectionStatus Struct Reference

### Data Fields

- uint8\_t [MDMConnStatus](#)
- uint64\_t [MDMCallDuration](#)

### 8.109.1 Detailed Description

#### Parameters

<i>MDMConn-Status</i>	Current link status 1-Disconnected 2-Connected
<i>MDMCall-Duration</i>	Call duration in milliseconds

### 8.109.2 Field Documentation

8.109.2.1 **uint64\_t** connectionStatus::MDMCallDuration

8.109.2.2 **uint8\_t** connectionStatus::MDMConnStatus

## 8.110 connectNumInfo Struct Reference

### Data Fields

- BYTE** numPresInd

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

### 8.110.1 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>

### 8.110.2 Field Documentation

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

8.110.2.2 **BYTE** connectNumInfo::callerIDLen

8.110.2.3 **BYTE** connectNumInfo::numPlan

8.110.2.4 **BYTE** connectNumInfo::numPresInd

8.110.2.5 **BYTE** connectNumInfo::numType

8.110.2.6 **BYTE** connectNumInfo::screeningInd

## 8.111 CrashInfo Struct Reference

### Data Fields

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

### 8.111.1 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>

## 8.111.2 Field Documentation

8.111.2.1 **ULONG** CrashInfo::crashData8.111.2.2 **ULONG** CrashInfo::crashId8.111.2.3 **WORD** CrashInfo::crashStrLen8.111.2.4 **WORD** CrashInfo::gcDumpStrLen8.111.2.5 **WORD** CrashInfo::numCrashes8.111.2.6 **CHAR\*** CrashInfo::pCrashString8.111.2.7 **CHAR\*** CrashInfo::pGCDumpString

## 8.112 CrashInfoParams Struct Reference

## Data Fields

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

## 8.112.1 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> (Optional parameter)</li> <li>• See <a href="#">CrashInfo</a> for more information</li> </ul>

## 8.112.2 Field Documentation

8.112.2.1 **CrashInfo\*** CrashInfoParams::pCrashInfo8.112.2.2 **BYTE\*** CrashInfoParams::pDevCrashStatus

## 8.113 CreateProfileIn Struct Reference

## Data Fields

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

## 8.113.1 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

## 8.113.2 Field Documentation

8.113.2.1 **QmiProfileInfo** CreateProfileIn::curProfile

8.113.2.2 **BYTE\*** CreateProfileIn::pProfileID

8.113.2.3 **BYTE\*** CreateProfileIn::pProfileType

## 8.114 CreateProfileOut Struct Reference

### Data Fields

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

### 8.114.1 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>

### 8.114.2 Field Documentation

8.114.2.1 **USHORT\*** CreateProfileOut::pExtErrorCode

8.114.2.2 **BYTE\*** CreateProfileOut::pProfileIndex

8.114.2.3 **BYTE\*** CreateProfileOut::pProfileType

## 8.115 CSGID Struct Reference

### Data Fields

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

### 8.115.1 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>

## 8.115.2 Field Documentation

8.115.2.1 ULONG CSGID::id

8.115.2.2 WORD CSGID::mcc

8.115.2.3 WORD CSGID::mnc

8.115.2.4 BYTE CSGID::mncPcsDigits

8.115.2.5 BYTE CSGID::rat

## 8.116 CUGInfo Struct Reference

## Data Fields

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

## 8.116.1 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>

## 8.116.2 Field Documentation

### 8.116.2.1 WORD CUGInfo::CUGIndex

### 8.116.2.2 BYTE CUGInfo::SuppOA

### 8.116.2.3 BYTE CUGInfo::SuppPrefCUG

## 8.117 curAMRConfig Struct Reference

### Data Fields

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

### 8.117.1 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>

## 8.117.2 Field Documentation

8.117.2.1 **BYTE** curAMRConfig::gsmAmrStat

8.117.2.2 **BYTE** curAMRConfig::wcdmaAmrStat

## 8.118 CurrDataSysStat Struct Reference

### Data Fields

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

### 8.118.1 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>

### 8.118.2 Field Documentation

8.118.2.1 **CurrNetworkInfo**\* CurrDataSysStat::pCurrNetworkInfo

8.118.2.2 **BYTE**\* CurrDataSysStat::pNetworkInfoLen

8.118.2.3 **BYTE**\* CurrDataSysStat::pPrefNetwork

## 8.119 currentCatEvent Union Reference

### Data Fields

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

### 8.119.1 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

### 8.119.2 Field Documentation

8.119.2.1 struct [CatAlPhalIdentifierTlv](#) currentCatEvent::CatAlphaldtfr

8.119.2.2 struct [CatEndProactiveSessionTlv](#) currentCatEvent::CatEndPS

8.119.2.3 struct [CatEventListTlv](#) currentCatEvent::CatEventLst

8.119.2.4 struct [CatEventIDDataTlv](#) currentCatEvent::CatEvIDData

8.119.2.5 struct [CatRefreshTlv](#) currentCatEvent::CatRefresh

## 8.120 CurrentImgList Struct Reference

### Data Fields

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

### 8.120.1 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 pCurrImgInfo 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>

## 8.120.2 Field Documentation

8.120.2.1 CHAR CurrentImgList::carrier[16]

8.120.2.2 CHAR CurrentImgList::fwvers[16]

8.120.2.3 BYTE CurrentImgList::numEntries

8.120.2.4 CurrImageInfo\* CurrentImgList::pCurrImgInfo

8.120.2.5 CHAR CurrentImgList::pkgver[16]

8.120.2.6 CHAR CurrentImgList::priver[16]

## 8.121 currentPLMN Struct Reference

## Data Fields

- WORD MCC
- WORD MNC
- BYTE netDescrLength
- BYTE netDescr [255]

## 8.121.1 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>

## 8.121.2 Field Documentation

8.121.2.1 WORD currentPLMN::MCC

8.121.2.2 WORD currentPLMN::MNC

8.121.2.3 BYTE currentPLMN::netDescr[255]

8.121.2.4 BYTE currentPLMN::netDescrLength

## 8.122 CurrImageInfo Struct Reference

## Data Fields

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

## 8.122.1 Detailed Description

This structure is used to store image information

## Parameters

<i>imageType[OUT]</i>	<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[OUT]</i>	<ul style="list-style-type: none"><li>• Image Unique Identifier ( ASCII characters )</li></ul>
<i>buildIDLen[OUT]</i>	<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[OUT]</i>	<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>

### 8.122.2 Field Documentation

8.122.2.1 **BYTE** CurrImageInfo::buildID[255]

8.122.2.2 **BYTE** CurrImageInfo::buildIDLen

8.122.2.3 **BYTE** CurrImageInfo::imageType

8.122.2.4 **BYTE** CurrImageInfo::uniqueID[16]

## 8.123 CurrNetworkInfo Struct Reference

### Data Fields

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

### 8.123.1 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> <li>• CDMA EV-DO Rev B 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> <li>– 0x10 - MMPA</li> <li>– 0x20 - MMPA_EHRPD</li> </ul> </li> </ul>
---------------	--

### 8.123.2 Field Documentation

8.123.2.1 **BYTE** CurrNetworkInfo::NetworkType

8.123.2.2 **ULONG** CurrNetworkInfo::RATMask

8.123.2.3 **ULONG** CurrNetworkInfo::SOMask

## 8.124 currNetworkInfo Struct Reference

### Data Fields

- uint8\_t [NetworkType](#)
- uint32\_t [RATMask](#)
- uint32\_t [SOMask](#)

### 8.124.1 Detailed Description

#### Parameters

<i>NetworkType</i>	network type 0-3GPP 1-3GPP2
<i>RATMask</i>	Radio access technology mask to indicate technology.
<i>SOMask</i>	Service Option mask to indicate the service option.

### 8.124.2 Field Documentation

8.124.2.1 uint8\_t currNetworkInfo::NetworkType

8.124.2.2 uint32\_t currNetworkInfo::RATMask

8.124.2.3 uint32\_t currNetworkInfo::SOMask

## 8.125 custFeaturesInfo Struct Reference

### Data Fields

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

### 8.125.1 Detailed Description

This structure contains current settings of custom features

#### Parameters

<i>GpsEnable[OUT]</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[OUT]</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[OUT]</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[OUT]</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[OUT]</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[-OUT]</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>pIsVoice-Enabled[OUT]</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[OUT]</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[OUT]</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>

### 8.125.2 Field Documentation

- 8.125.2.1 **ULONG** custFeaturesInfo::GpsEnable
- 8.125.2.2 **BYTE\*** custFeaturesInfo::pDHCPRelayEnabled
- 8.125.2.3 **BYTE\*** custFeaturesInfo::pDisableIMSI
- 8.125.2.4 **BYTE\*** custFeaturesInfo::pGPSLPM
- 8.125.2.5 **BYTE\*** custFeaturesInfo::pGPSSel
- 8.125.2.6 **WORD\*** custFeaturesInfo::pIPFamSupport
- 8.125.2.7 **BYTE\*** custFeaturesInfo::plsVoiceEnabled
- 8.125.2.8 **BYTE\*** custFeaturesInfo::pRMAutoConnect
- 8.125.2.9 **BYTE\*** custFeaturesInfo::pSMSSupport

## 8.126 custFeaturesSetting Struct Reference

### Data Fields

- [BYTE \\*](#) [pGPSSel](#)
- [ULONG \\*](#) [pGPSEnable](#)
- [BYTE \\*](#) [plsVoiceEnabled](#)
- [BYTE \\*](#) [pDHCPRelayEnabled](#)
- [BYTE \\*](#) [pGPSLPM](#)

### 8.126.1 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>

## 8.126.2 Field Documentation

8.126.2.1 **BYTE\*** custFeaturesSetting::pDHCPRelayEnabled

8.126.2.2 **ULONG\*** custFeaturesSetting::pGPSEnable

8.126.2.3 **BYTE\*** custFeaturesSetting::pGPSLPM

8.126.2.4 **BYTE\*** custFeaturesSetting::pGPSSel

8.126.2.5 **BYTE\*** custFeaturesSetting::plsVoiceEnabled

## 8.127 custSettingInfo Struct Reference

### Data Fields

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

### 8.127.1 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>

## 8.127.2 Field Documentation

8.127.2.1 **WORD** custSettingInfo::cust\_attr

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

8.127.2.3 **BYTE** `custSettingInfo::cust_value[8+1]`

8.127.2.4 **WORD** `custSettingInfo::id_length`

8.127.2.5 **WORD** `custSettingInfo::value_length`

## 8.128 `custSettingList` Struct Reference

### Data Fields

- [BYTE](#) `list_type`
- [WORD](#) `num_instances`
- [custSettingInfo](#) `custSetting` [256]

### 8.128.1 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>

### 8.128.2 Field Documentation

8.128.2.1 **custSettingInfo** `custSettingList::custSetting[256]`

8.128.2.2 **BYTE** `custSettingList::list_type`

8.128.2.3 **WORD** `custSettingList::num_instances`

## 8.129 `dataBearers` Struct Reference

### Data Fields

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

### 8.129.1 Detailed Description

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>

## 8.129.2 Field Documentation

8.129.2.1 BYTE `dataBearers::dataBearerMask`8.129.2.2 QmiWSDDataBearerTechnology\* `dataBearers::pCurDataBearerTechnology`8.129.2.3 QmiWSDDataBearerTechnology\* `dataBearers::pLastCallDataBearerTechnology`

## 8.130 DataBearerTech Struct Reference

## Data Fields

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

## 8.130.1 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> <li>• 3GPP2 SO mask: <ul style="list-style-type: none"> <li>– 0x01000000 - 1X IS95</li> <li>– 0x02000000 - 1X IS2000</li> <li>– 0x04000000 - 1X IS2000 REL A</li> <li>– 0x08000000 - HDR REV0 DPA</li> <li>– 0x10000000 - HDR REVA DPA</li> <li>– 0x20000000 - HDR REVB DPA</li> <li>– 0x40000000 - HDR REVA MPA</li> <li>– 0x80000000 - HDR REVB MPA</li> <li>– 0x100000000 - HDR REVA EMPA</li> <li>– 0x200000000 - HDR REVB EMPA</li> <li>– 0x400000000 - HDR REVB MMPA</li> <li>– 0x800000000 - HDR EVDO FMC</li> </ul> </li> </ul>
---------------	---

## 8.130.2 Field Documentation

8.130.2.1 **ULONG** DataBearerTech::ratValue

8.130.2.2 **ULONGLONG** DataBearerTech::soMask

8.130.2.3 **ULONG** DataBearerTech::techType

## 8.131 DataBearerTechExt Struct Reference

### Data Fields

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

### 8.131.1 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>

### 8.131.2 Field Documentation

8.131.2.1 **DataBearerTech\*** DataBearerTechExt::pBearerTech

8.131.2.2 **DataBearerTech\*** DataBearerTechExt::pLastBearerTech

## 8.132 dataBearerTechnology Struct Reference

### Data Fields

- [BYTE](#) currentNetwork
- [ULONG](#) ratMask
- [ULONG](#) soMask

### 8.132.1 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>

## 8.132.2 Field Documentation

8.132.2.1 **BYTE** dataBearerTechnology::currentNetwork

8.132.2.2 **ULONG** dataBearerTechnology::ratMask

8.132.2.3 **ULONG** dataBearerTechnology::soMask

## 8.133 dataRate Struct Reference

### Data Fields

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

### 8.133.1 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)

## 8.133.2 Field Documentation

8.133.2.1 **ULONG** dataRate::dataRateMax8.133.2.2 **ULONG** dataRate::guaranteedRate

## 8.134 dataSrvCapabilities Struct Reference

## Data Fields

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

## 8.134.1 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>

## 8.134.2 Field Documentation

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

8.134.2.2 BYTE dataSrvCapabilities::dataCapabilitiesLen

## 8.135 DataStatusDetail Struct Reference

### Data Fields

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

### 8.135.1 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 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> <li>– 0x8F - MIP_RRP_CODE_FAILURE_REDIRECTED_HA</li> <li>– 0x90 - MIP_RRP_CODE_FAILURE_HA_BAD_AAA_AUTH</li> <li>– 0xFF - NA</li> </ul> </li> </ul>
--------------------	--

## 8.135.2 Field Documentation

### 8.135.2.1 ULONG DataStatusDetail::IPAddress

### 8.135.2.2 BYTE DataStatusDetail::LastErrCode

## 8.136 DataULongLongTlv Struct Reference

### Data Fields

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

#### 8.136.1 Field Documentation

8.136.1.1 **BYTE** DataULongLongTlv::TlvPresent

8.136.1.2 **ULONGLONG** DataULongLongTlv::ullData

## 8.137 DataULongTlv Struct Reference

### Data Fields

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

#### 8.137.1 Field Documentation

8.137.1.1 **BYTE** DataULongTlv::TlvPresent

8.137.1.2 **ULONG** DataULongTlv::ulData

## 8.138 DcsUsbPortNames Struct Reference

### Data Fields

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

#### 8.138.1 Field Documentation

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

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

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

## 8.139 delAssistDataStatus Struct Reference

### Data Fields

- [ULONG status](#)

### 8.139.1 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

### 8.139.2 Field Documentation

#### 8.139.2.1 ULONG delAssistDataStatus::status

## 8.140 depersonalizationInformation Struct Reference

#### Data Fields

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

### 8.140.1 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_DE- SCRIPTION_LE- NGTH]	<ul style="list-style-type: none"> <li>Control key value.</li> <li>This value is a sequence of ASCII characters.</li> </ul>

## 8.140.2 Field Documentation

8.140.2.1 BYTE depersonalizationInformation::ckLen

8.140.2.2 BYTE depersonalizationInformation::ckVal[255]

8.140.2.3 BYTE depersonalizationInformation::feature

8.140.2.4 BYTE depersonalizationInformation::operation

## 8.141 detailSvcInfo Struct Reference

## Data Fields

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

## 8.141.1 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>

## 8.141.2 Field Documentation

8.141.2.1 BYTE detailSvcInfo::hdrHybrid

8.141.2.2 BYTE detailSvcInfo::hdrSrvStatus

8.141.2.3 BYTE detailSvcInfo::isSysForbidden

8.141.2.4 BYTE detailSvcInfo::srvCapability

8.141.2.5 BYTE detailSvcInfo::srvStatus

## 8.142 DeviceConfigDetail Struct Reference

### Data Fields

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

### 8.142.1 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>

### 8.142.2 Field Documentation

8.142.2.1 BYTE DeviceConfigDetail::Chipset

8.142.2.2 **BYTE** DeviceConfigDetail::HWVersion

8.142.2.3 **BYTE** DeviceConfigDetail::QLIC

8.142.2.4 **BYTE** DeviceConfigDetail::Technology

## 8.143 DHCPOption Struct Reference

### Data Fields

- [BYTE optCode](#)
- [BYTE optValLen](#)
- [BYTE \\* pOptVal](#)

### 8.143.1 Detailed Description

This structure contains DHCPv4 lease option values

#### Parameters

<i>optCode</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Option code</li> </ul> </li> </ul>
<i>optValLen</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Option value length</li> </ul> </li> </ul>
<i>pOptValue</i>	<ul style="list-style-type: none"> <li>• Val <ul style="list-style-type: none"> <li>– Option value</li> </ul> </li> </ul>

### 8.143.2 Field Documentation

8.143.2.1 **BYTE** DHCPOption::optCode

8.143.2.2 **BYTE** DHCPOption::optValLen

8.143.2.3 **BYTE\*** DHCPOption::pOptVal

## 8.144 DHCPOptionList Struct Reference

### Data Fields

- [BYTE numOpt](#)
- [DHCPOption \\* pOptions](#)

### 8.144.1 Detailed Description

This structure contains DHCPv4 lease option list

## Parameters

<i>optListSize</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Size of Option List</li> </ul> </li> </ul>
<i>pOptions</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– Options</li> </ul> </li> </ul>

## 8.144.2 Field Documentation

8.144.2.1 BYTE DHCPOptionList::numOpt

8.144.2.2 DHCPOption\* DHCPOptionList::pOptions

## 8.145 diagInfo Struct Reference

## Data Fields

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

## 8.145.1 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_DESCRIPTI-ON_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Diagnostic information.</li> </ul>

## 8.145.2 Field Documentation

8.145.2.1 BYTE diagInfo::diagInfoLen

8.145.2.2 BYTE diagInfo::diagnosticInfo[255]

## 8.146 dirNum Struct Reference

## Data Fields

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

### 8.146.1 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><i>dir_num</i></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>

### 8.146.2 Field Documentation

8.146.2.1 **BYTE** *dirNum::dirNum*[255]

8.146.2.2 **BYTE** *dirNum::dirNumLen*

## 8.147 dms\_ActivationStatusTlv Struct Reference

### Data Fields

- uint16\_t [TlvPresent](#)
- uint32\_t [activationStatus](#)

### 8.147.1 Detailed Description

Activation Status Tlv

#### 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>
-------------------------	--

### 8.147.2 Field Documentation

8.147.2.1 **uint32\_t** *dms\_ActivationStatusTlv::activationStatus*

8.147.2.2 uint16\_t dms\_ActivationStatusTlv::TlvPresent

## 8.148 dms\_OperatingModeTlv Struct Reference

### Data Fields

- uint16\_t [TlvPresent](#)
- uint32\_t [operatingMode](#)

### 8.148.1 Detailed Description

Operating Mode Tlv

#### Parameters

<i>operatingMode</i>	<ul style="list-style-type: none"> <li>• 0 - Online</li> <li>• 1 - Low power</li> <li>• 2 - Factory test mode</li> <li>• 3 - Offline</li> <li>• 4 - Resetting</li> <li>• 5 - Shutting down</li> <li>• 6 - Persistent low power</li> <li>• 7 - Mode-only low power</li> </ul>
----------------------	--

### 8.148.2 Field Documentation

8.148.2.1 uint32\_t dms\_OperatingModeTlv::operatingMode

8.148.2.2 uint16\_t dms\_OperatingModeTlv::TlvPresent

## 8.149 dmsCurrentPRLInfo Struct Reference

### Data Fields

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

### 8.149.1 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>

## 8.149.2 Field Documentation

8.149.2.1 **BYTE\*** dmsCurrentPRLInfo::pPRLPreference

8.149.2.2 **WORD\*** dmsCurrentPRLInfo::pPRLVersion

## 8.150 DMScustSettingInfo Struct Reference

### Data Fields

- uint16\_t [id\\_length](#)
- uint8\_t [cust\\_id](#) [64+1]
- uint16\_t [value\\_length](#)
- uint8\_t [cust\\_value](#) [8+1]
- uint16\_t [cust\\_attr](#)

### 8.150.1 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>

## 8.150.2 Field Documentation

8.150.2.1 **uint16\_t** DMScustSettingInfo::cust\_attr

8.150.2.2 **uint8\_t** DMScustSettingInfo::cust\_id[64+1]

8.150.2.3 **uint8\_t** DMScustSettingInfo::cust\_value[8+1]

8.150.2.4 **uint16\_t** DMScustSettingInfo::id\_length

8.150.2.5 **uint16\_t** DMScustSettingInfo::value\_length

## 8.151 DMScustSettingList Struct Reference

### Data Fields

- [uint8\\_t list\\_type](#)
- [uint16\\_t num\\_instances](#)
- [DMScustSettingInfo custSetting](#) [255+1]

### 8.151.1 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>

### 8.151.2 Field Documentation

8.151.2.1 [DMScustSettingInfo](#) [DMScustSettingList::custSetting](#)[255+1]

8.151.2.2 [uint8\\_t](#) [DMScustSettingList::list\\_type](#)

8.151.2.3 [uint16\\_t](#) [DMScustSettingList::num\\_instances](#)

## 8.152 DMSgetCustomFeatureV2 Struct Reference

### Data Fields

- [DMSgetCustomInput](#) \* [pGetCustomInput](#)
- [DMScustSettingInfo](#) \* [pCustSettingInfo](#)
- [DMScustSettingList](#) \* [pCustSettingList](#)

### 8.152.1 Detailed Description

This struture contains the TLV required to get the Customization Info and customization list.

#### Parameters

<i>IN]</i>	<a href="#">pGetCustomInput</a> <ul style="list-style-type: none"><li>• See <a href="#">getCustomInput</a> for more information</li></ul>
<i>OUT]</i>	<a href="#">pCustSettingInfo</a> <ul style="list-style-type: none"><li>• See <a href="#">custSettingInfo</a> for more information</li></ul>

<i>OUT]</i>	<p>pCustSettingList</p> <ul style="list-style-type: none"> <li>• See <a href="#">custSettingList</a> for more information</li> </ul>
-------------	--

### 8.152.2 Field Documentation

8.152.2.1 DMScustSettingInfo\* DMSgetCustomFeatureV2::pCustSettingInfo

8.152.2.2 DMScustSettingList\* DMSgetCustomFeatureV2::pCustSettingList

8.152.2.3 DMSgetCustomInput\* DMSgetCustomFeatureV2::pGetCustomInput

## 8.153 DMSgetCustomInput Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint8\_t [list\\_type](#)

### 8.153.1 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>

### 8.153.2 Field Documentation

8.153.2.1 uint8\_t DMSgetCustomInput::cust\_id[64+1]

8.153.2.2 uint8\_t DMSgetCustomInput::list\_type

## 8.154 dmsIndicationRegisterReq Struct Reference

### Data Fields

- [BYTE](#) \* [pSwiGetResetInd](#)

### 8.154.1 Detailed Description

This structure contains the SLQSDmsSwiIndicationRegister request parameters.

## Parameters

<i>IN</i> ]	<p>pGetResetInd [Optional]</p> <ul style="list-style-type: none"> <li>• Get Reset Info indication registration. The following callbacks would not be invoked if the indication is disabled. <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
-------------	---

## Note

'NULL' value confirms that the indication value is not sent.

## 8.154.2 Field Documentation

8.154.2.1 **BYTE\*** dmsIndicationRegisterReq::pSwiGetResetInd

## 8.155 dmsSwiGetResetInfo Struct Reference

## Data Fields

- [BYTE type](#)
- [BYTE source](#)

## 8.155.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

## Parameters

<i>OUT</i> ]	<p>type[OUT]</p> <ul style="list-style-type: none"> <li>• type of reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - warm</li> <li>– 2 - hard</li> <li>– 3 - crash</li> <li>– 4 - power down</li> </ul> </li> </ul>
<i>OUT</i> ]	<p>source[OUT]</p> <ul style="list-style-type: none"> <li>• entity which initiated the reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - user requested</li> <li>– 2 - hardware switch</li> <li>– 3 - temperature critical</li> <li>– 4 - voltage critical</li> <li>– 5 - configuration update</li> <li>– 6 - LWM2M</li> <li>– 7 - OMA-DM</li> <li>– 8 - FOTA</li> </ul> </li> </ul>

## 8.155.2 Field Documentation

8.155.2.1 **BYTE** dmsSwiGetResetInfo::source

8.155.2.2 **BYTE** dmsSwiGetResetInfo::type

## 8.156 Domain Struct Reference

### Data Fields

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

### 8.156.1 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>

### 8.156.2 Field Documentation

8.156.2.1 **WORD** Domain::domainLen

8.156.2.2 **CHAR** Domain::domainName[256]

## 8.157 DomainNameList Struct Reference

### Data Fields

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

### 8.157.1 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>

### 8.157.2 Field Documentation

8.157.2.1 struct Domain DomainNameList::domain[10]

8.157.2.2 BYTE DomainNameList::numInstances

## 8.158 DRCPParams Struct Reference

### Data Fields

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

### 8.158.1 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>

### 8.158.2 Field Documentation

8.158.2.1 BYTE DRCPParams::DRCCover

8.158.2.2 BYTE DRCPParams::DRCValue

## 8.159 DTMFInfo Struct Reference

### Data Fields

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

### 8.159.1 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[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• DTMF digit buffer in ASCII string which is NULL terminated</li> </ul>

## 8.159.2 Field Documentation

8.159.2.1 BYTE DTMFInfo::callID

8.159.2.2 BYTE DTMFInfo::digitBuff[255]

8.159.2.3 BYTE DTMFInfo::digitCnt

8.159.2.4 BYTE DTMFInfo::DTMFEvent

## 8.160 DTMFLengths Struct Reference

## Data Fields

- [BYTE DTMFPulseWidth](#)
- [BYTE DTMFInterdigitInterval](#)

## 8.160.1 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>

## 8.160.2 Field Documentation

8.160.2.1 BYTE DTMFLengths::DTMFInterdigitInterval

8.160.2.2 BYTE DTMFLengths::DTMFPulseWidth

## 8.161 DUNCallInfoInd Struct Reference

## Data Fields

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

## 8.161.1 Field Documentation

8.161.1.1 WORD DUNCallInfoInd::CallEndReason

8.161.1.2 channelRate DUNCallInfoInd::ChannelRate

8.161.1.3 BYTE DUNCallInfoInd::DataBearerTech

8.161.1.4 BYTE DUNCallInfoInd::DormancyStatus

8.161.1.5 BYTE DUNCallInfoInd::MdmConnStatus

8.161.1.6 ULONGLONG DUNCallInfoInd::RXOKBytesCount

8.161.1.7 ULONGLONG DUNCallInfoInd::TXOKBytesCount

## 8.162 dunchannelRate Struct Reference

### Data Fields

- uint32\_t [CurrChanTxRate](#)
- uint32\_t [CurrChanRxRate](#)
- uint32\_t [MaxChanTxRate](#)
- uint32\_t [MaxChanRxRate](#)

### 8.162.1 Detailed Description

#### Parameters

<i>CurrChanTxRate</i>	instantaneous channel Tx rate in bits per second
<i>CurrChanRxRate</i>	instantaneous channel Rx rate in bits per second
<i>MaxChanTxRate</i>	maximum Tx rate that can be assigned to the device
<i>MaxChanRxRate</i>	maximum Rx rate that can be assigned to the device

### 8.162.2 Field Documentation

8.162.2.1 uint32\_t dunchannelRate::CurrChanRxRate

8.162.2.2 uint32\_t dunchannelRate::CurrChanTxRate

8.162.2.3 uint32\_t dunchannelRate::MaxChanRxRate

8.162.2.4 uint32\_t dunchannelRate::MaxChanTxRate

## 8.163 ecioListElement Struct Reference

### Data Fields

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

### 8.163.1 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>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_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>

## 8.163.2 Field Documentation

## 8.163.2.1 SHORT ecioListElement::ecio

## 8.163.2.2 BYTE ecioListElement::radioIf

## 8.164 ECIOThresh Struct Reference

## Data Fields

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

## 8.164.1 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>

## 8.164.2 Field Documentation

## 8.164.2.1 BYTE ECIOThresh::ECIOThresListLen

8.164.2.2 **SHORT\*** `ECIOThresh::pECIOThresList`

## 8.165 ECTNum Struct Reference

### Data Fields

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

### 8.165.1 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 - <code>ECT_CALL_STATE_NONE</code> - None</li> <li>0x01 - <code>ECT_CALL_STATE_ALERTING</code> - Alerting</li> <li>0x02 - <code>ECT_CALL_STATE_ACTIVE</code> - 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 - <code>presentationAllowedAddress</code></li> <li>0x01 - <code>presentationRestricted</code></li> <li>0x02 - <code>numberNotAvailable</code></li> <li>0x04 - <code>presentationRestrictedAddress</code></li> </ul> </li> </ul>
<i>number</i>	<ul style="list-style-type: none"> <li>Number in ASCII characters terminated by NULL</li> </ul>

### 8.165.2 Field Documentation

8.165.2.1 **BYTE** `ECTNum::ECTCallState`

8.165.2.2 **BYTE** `ECTNum::number`[81]

8.165.2.3 **BYTE** `ECTNum::presentationInd`

## 8.166 encryptedPIN1 Struct Reference

### Data Fields

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

### 8.166.1 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.

## 8.166.2 Field Documentation

8.166.2.1 BYTE encryptedPIN1::pin1Len

8.166.2.2 BYTE encryptedPIN1::pin1Val[255]

## 8.167 ERIFileparams Struct Reference

## Data Fields

- WORD \* pFileSize
- BYTE \* pFile

## 8.167.1 Detailed Description

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>

## 8.167.2 Field Documentation

8.167.2.1 BYTE\* ERIFileparams::pFile

8.167.2.2 WORD\* ERIFileparams::pFileSize

## 8.168 errorRateListElement Struct Reference

## Data Fields

- USHORT errorRate
- BYTE radiolf

### 8.168.1 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>

### 8.168.2 Field Documentation

8.168.2.1 USHORT errorRateListElement::errorRate

8.168.2.2 BYTE errorRateListElement::radioIf

## 8.169 eTWSPLMNInfoTlv Struct Reference

#### Data Fields

- uint8\_t TlvPresent
- sMSEtwsPlmnInfo ETWSPLMNInfo

### 8.169.1 Detailed Description

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>ETWSPLMNInfo</i>	<ul style="list-style-type: none"> <li>• ETWS PLMN Information</li> <li>• See <a href="#">sMSEtwsPlmnInfo</a> for more information</li> </ul>

## 8.169.2 Field Documentation

8.169.2.1 [sMSEtwsPlmnInfo](#) `eTWSPLMNInfoTlv::ETWSPLMNInfo`8.169.2.2 `uint8_t eTWSPLMNInfoTlv::TlvPresent`

## 8.170 extDispRecInfo Struct Reference

## Data Fields

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

## 8.170.1 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>– <code>ext_display_info</code></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>

## 8.170.2 Field Documentation

8.170.2.1 `BYTE extDispRecInfo::dispType`8.170.2.2 `BYTE extDispRecInfo::extDispInfo[255]`8.170.2.3 `BYTE extDispRecInfo::extDispInfoLen`

## 8.171 FactorySequenceNumber Struct Reference

## Data Fields

- [BYTE FSNumber](#) [255]

### 8.171.1 Detailed Description

This structure used to store Factory Sequence Number parameter

#### Parameters

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

### 8.171.2 Field Documentation

#### 8.171.2.1 BYTE FactorySequenceNumber::FSNumber[255]

## 8.172 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]

### 8.172.1 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>secDeactivate-Mask</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>
--	---

## 8.172.2 Field Documentation

8.172.2.1 WORD fileAttributes::fileID

8.172.2.2 WORD fileAttributes::fileSize

8.172.2.3 BYTE fileAttributes::fileType

8.172.2.4 WORD fileAttributes::rawLen

8.172.2.5 BYTE fileAttributes::rawValue[255]

8.172.2.6 WORD fileAttributes::recordCount

8.172.2.7 WORD fileAttributes::recordSize

8.172.2.8 BYTE fileAttributes::secActivate

8.172.2.9 WORD fileAttributes::secActivateMask

8.172.2.10 BYTE fileAttributes::secDeactivate

8.172.2.11 WORD fileAttributes::secDeactivateMask

8.172.2.12 BYTE fileAttributes::secIncrease

8.172.2.13 WORD fileAttributes::secIncreaseMask

8.172.2.14 BYTE fileAttributes::secRead

8.172.2.15 WORD fileAttributes::secReadMask

8.172.2.16 BYTE fileAttributes::secWrite

8.172.2.17 WORD fileAttributes::secWriteMask

## 8.173 fileInfo Struct Reference

### Data Fields

- WORD fileID
- BYTE pathLen
- WORD path [255]

### 8.173.1 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>

## 8.173.2 Field Documentation

### 8.173.2.1 WORD fileInfo::fileID

### 8.173.2.2 WORD fileInfo::path[255]

### 8.173.2.3 BYTE fileInfo::pathLen

## 8.174 FirmwareUpdatStat Struct Reference

### Data Fields

- [ULONG ResCode](#)
- [BYTE \\* plmgType](#)
- [ULONG \\* pRefData](#)
- [BYTE \\* pRefStringLen](#)
- [BYTE \\* pRefString](#)
- [BYTE \\* pLogStringLen](#)
- [BYTE \\* pLogString](#)

### 8.174.1 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>0x40000nnn - 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>Optional parameter</li> <li>Firmware image type that failed the update</li> </ul>
<i>pRefData</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <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>Optional parameter</li> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>
<i>pLogStringLen</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>pLogString</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>

## 8.174.2 Field Documentation

## 8.174.2.1 BYTE\* FirmwareUpdatStat::plmgType

8.174.2.2 **BYTE\*** FirmwareUpdatStat::pLogString

8.174.2.3 **BYTE\*** FirmwareUpdatStat::pLogStringLen

8.174.2.4 **ULONG\*** FirmwareUpdatStat::pRefData

8.174.2.5 **BYTE\*** FirmwareUpdatStat::pRefString

8.174.2.6 **BYTE\*** FirmwareUpdatStat::pRefStringLen

8.174.2.7 **ULONG** FirmwareUpdatStat::ResCode

## 8.175 FMSImageElement Struct Reference

### Data Fields

- uint8\_t [imageType](#)
- uint8\_t [imageId](#) [16]
- uint8\_t [buildIdLength](#)
- uint8\_t [buildId](#) [100]

### 8.175.1 Detailed Description

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>

### 8.175.2 Field Documentation

8.175.2.1 uint8\_t FMSImageElement::buildId[100]

8.175.2.2 uint8\_t FMSImageElement::buildIdLength

8.175.2.3 uint8\_t FMSImageElement::imageId[16]

8.175.2.4 uint8\_t FMSImageElement::imageType

## 8.176 FMSImageIdElement Struct Reference

## Data Fields

- uint8\_t [storageIndex](#)
- uint8\_t [failureCount](#)
- uint8\_t [imageID](#) [16]
- uint8\_t [buildIDLength](#)
- uint8\_t [buildID](#) [100]

### 8.176.1 Detailed Description

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>

### 8.176.2 Field Documentation

8.176.2.1 uint8\_t FMSImageIDElement::buildID[100]

8.176.2.2 uint8\_t FMSImageIDElement::buildIDLength

8.176.2.3 uint8\_t FMSImageIDElement::failureCount

8.176.2.4 uint8\_t FMSImageIDElement::imageID[16]

8.176.2.5 uint8\_t FMSImageIDElement::storageIndex

## 8.177 FMSImageIDEntries Struct Reference

## Data Fields

- uint8\_t [imageType](#)
- uint8\_t [maxImages](#)
- uint8\_t [executingImage](#)
- uint8\_t [imageIDSize](#)
- [FMSImageIDElement](#) [imageIDElement](#) [50]

### 8.177.1 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> <li>See <a href="#">FMSImageIDElement</a></li> </ul>

### 8.177.2 Field Documentation

8.177.2.1 `uint8_t FMSImageIDEntries::executingImage`

8.177.2.2 `FMSImageIDElement FMSImageIDEntries::imageIDElement[50]`

8.177.2.3 `uint8_t FMSImageIDEntries::imageIDSize`

8.177.2.4 `uint8_t FMSImageIDEntries::imageType`

8.177.2.5 `uint8_t FMSImageIDEntries::maxImages`

## 8.178 FMSImageList Struct Reference

### Data Fields

- `uint8_t listSize`
- `FMSImageIDEntries imageIDEntries [2]`

### 8.178.1 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>
-----------------------	---

## 8.178.2 Field Documentation

8.178.2.1 **FMSImageIDEntries** FMSImageList::imageIDEntries[2]

8.178.2.2 **uint8\_t** FMSImageList::listSize

## 8.179 FMSPrefImageList Struct Reference

### Data Fields

- [uint8\\_t listSize](#)
- [FMSImageElement listEntries](#) [2]

### 8.179.1 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="#">FMSImageElement</a></li> </ul>

## 8.179.2 Field Documentation

8.179.2.1 **FMSImageElement** FMSPrefImageList::listEntries[2]

8.179.2.2 **uint8\_t** FMSPrefImageList::listSize

## 8.180 fwinfo\_s Struct Reference

### Data Fields

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

### 8.180.1 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>

## 8.180.2 Field Documentation

8.180.2.1 **ULONG** fwinfo\_s::Carrier8.180.2.2 **ULONG** fwinfo\_s::FirmwareID8.180.2.3 **ULONG** fwinfo\_s::GPSCapability8.180.2.4 **ULONG** fwinfo\_s::Region8.180.2.5 **ULONG** fwinfo\_s::Technology

## 8.181 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]

## 8.181.1 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</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 <i>nmrCellID</i> 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 <i>nmrCellID</i> 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</i> [ <i>MAX_DESCRIPTION_LENGTH</i> ]	<ul style="list-style-type: none"> <li>• See <a href="#">nmrCellInfo</a> for more information.</li> </ul>

## 8.181.2 Field Documentation

### 8.181.2.1 WORD GERANInfo::arfcn

### 8.181.2.2 BYTE GERANInfo::bsic

### 8.181.2.3 ULONG GERANInfo::cellID

8.181.2.4 **nmrCellInfo** GERANInfo::insNmrCellInfo[255]

8.181.2.5 **WORD** GERANInfo::lac

8.181.2.6 **BYTE** GERANInfo::nmrInst

8.181.2.7 **BYTE** GERANInfo::plmn[3]

8.181.2.8 **WORD** GERANInfo::rxLev

8.181.2.9 **ULONG** GERANInfo::timingAdvance

## 8.182 geranInstInfo Struct Reference

### Data Fields

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

### 8.182.1 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>

### 8.182.2 Field Documentation

8.182.2.1 **WORD** `geranInstInfo::geranArfcn`

8.182.2.2 **BYTE** `geranInstInfo::geranBsicBcc`

8.182.2.3 **BYTE** `geranInstInfo::geranBsicNcc`

8.182.2.4 **SHORT** `geranInstInfo::geranRssi`

## 8.183 getAllCallInformation Struct Reference

### Data Fields

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

### 8.183.1 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>

### 8.183.2 Field Documentation

8.183.2.1 [BYTE](#) [getAllCallInformation::ALS](#)

8.183.2.2 [callInfo](#) [getAllCallInformation::Callinfo](#)

8.183.2.3 [BYTE](#) [getAllCallInformation::isEmpty](#)

## 8.184 getAllCallRmtPtyName Struct Reference

### Data Fields

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

### 8.184.1 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>
-------------------------	---

### 8.184.2 Field Documentation

8.184.2.1 **BYTE** getAllCallRmtPtyName::callID

8.184.2.2 **remotePartyName** getAllCallRmtPtyName::RemotePartyName

## 8.185 getAllCallRmtPtyNum Struct Reference

### Data Fields

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

### 8.185.1 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>

### 8.185.2 Field Documentation

8.185.2.1 **BYTE** getAllCallRmtPtyNum::callID

8.185.2.2 **remotePartyNum** getAllCallRmtPtyNum::RemotePartyNum

## 8.186 GetAudioPathConfigReq Struct Reference

### Data Fields

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

### 8.186.1 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>

## 8.186.2 Field Documentation

8.186.2.1 BYTE GetAudioPathConfigReq::Item

8.186.2.2 BYTE GetAudioPathConfigReq::Profile

## 8.187 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](#)

## 8.187.1 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>pTXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">TXPCMIIRFiltr</a> for more information</li> </ul>
<i>pRXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">RXPCMIIRFiltr</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>

## 8.187.2 Field Documentation

- 8.187.2.1 **WORD\*** GetAudioPathConfigResp::pCodecSTGain
- 8.187.2.2 **WORD\*** GetAudioPathConfigResp::pDTMFTXGain
- 8.187.2.3 **BYTE\*** GetAudioPathConfigResp::pECMode
- 8.187.2.4 **BYTE\*** GetAudioPathConfigResp::pMICGainSelect
- 8.187.2.5 **BYTE\*** GetAudioPathConfigResp::pNSEnable
- 8.187.2.6 **RXAGCList\*** GetAudioPathConfigResp::pRXAGCList
- 8.187.2.7 **BYTE\*** GetAudioPathConfigResp::pRXAVCAGCSwitch
- 8.187.2.8 **RXAVCList\*** GetAudioPathConfigResp::pRXAVCList
- 8.187.2.9 **RXPCMIIRFiltr\*** GetAudioPathConfigResp::pRXPCMIIRFiltr
- 8.187.2.10 **TXAGCList\*** GetAudioPathConfigResp::pTXAGCList
- 8.187.2.11 **BYTE\*** GetAudioPathConfigResp::pTXAVCSwitch
- 8.187.2.12 **WORD\*** GetAudioPathConfigResp::pTXGain
- 8.187.2.13 **TXPCMIIRFiltr\*** GetAudioPathConfigResp::pTXPCMIIRFiltr

## 8.188 GetAudioProfileReq Struct Reference

### Data Fields

- [BYTE Generator](#)

### 8.188.1 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>
------------------	--

### 8.188.2 Field Documentation

- 8.188.2.1 **BYTE** GetAudioProfileReq::Generator

## 8.189 GetAudioProfileResp Struct Reference

## Data Fields

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

### 8.189.1 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>

### 8.189.2 Field Documentation

8.189.2.1 **BYTE** GetAudioProfileResp::EarMute

8.189.2.2 **BYTE** GetAudioProfileResp::MicMute

8.189.2.3 **BYTE** GetAudioProfileResp::Profile

8.189.2.4 **BYTE** GetAudioProfileResp::Volume

## 8.190 GetAudioVolTLBConfigReq Struct Reference

### Data Fields

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

### 8.190.1 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>

### 8.190.2 Field Documentation

8.190.2.1 **BYTE** GetAudioVolTLBConfigReq::Generator

8.190.2.2 **BYTE** GetAudioVolTLBConfigReq::Item

8.190.2.3 **BYTE** GetAudioVolTLBConfigReq::Profile

8.190.2.4 **BYTE** GetAudioVolTLBConfigReq::Volume

## 8.191 GetAudioVolTLBConfigResp Struct Reference

### Data Fields

- [WORD ResCode](#)

### 8.191.1 Detailed Description

This structure contains the SLQSGetAudioVoTLBConfig response parameters.

#### Parameters

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

### 8.191.2 Field Documentation

#### 8.191.2.1 WORD GetAudioVoTLBConfigResp::ResCode

## 8.192 getCallFWExtInfo Struct Reference

### Data Fields

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

### 8.192.1 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>

### 8.192.2 Field Documentation

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

#### 8.192.2.2 BYTE getCallFWExtInfo::numInstances

## 8.193 getCallFWInfo Struct Reference

### Data Fields

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

### 8.193.1 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[MAX_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>

## 8.193.2 Field Documentation

8.193.2.1 [callFWInfo](#) getCallFWInfo::CallFWInfo[20]

8.193.2.2 BYTE getCallFWInfo::numInstances

## 8.194 getCustomFeatureV2 Struct Reference

## Data Fields

- [getCustomInput](#) \* [pGetCustomInput](#)
- [custSettingInfo](#) \* [pCustSettingInfo](#)
- [custSettingList](#) \* [pCustSettingList](#)

## 8.194.1 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>[IN]</p> <ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• See <a href="#">getCustomInput</a> for more information</li> </ul>
<i>OUT]</i>	<p><a href="#">pCustSettingInfo</a>[OUT]</p> <ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• See <a href="#">custSettingInfo</a> for more information</li> </ul>
<i>OUT]</i>	<p><a href="#">pCustSettingList</a>[OUT]</p> <ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• See <a href="#">custSettingList</a> for more information</li> </ul>

## 8.194.2 Field Documentation

8.194.2.1 [custSettingInfo](#)\* getCustomFeatureV2::pCustSettingInfo8.194.2.2 [custSettingList](#)\* getCustomFeatureV2::pCustSettingList8.194.2.3 [getCustomInput](#)\* getCustomFeatureV2::pGetCustomInput

## 8.195 getCustomInput Struct Reference

### Data Fields

- [CHAR cust\\_id](#) [64+1]
- [BYTE list\\_type](#)

### 8.195.1 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>

### 8.195.2 Field Documentation

8.195.2.1 **CHAR** getCustomInput::cust\_id[64+1]

8.195.2.2 **BYTE** getCustomInput::list\_type

## 8.196 getDUNCallInfoReq Struct Reference

### Data Fields

- [ULONG Mask](#)
- [BYTE \\* pReportConnStatus](#)
- [TransferStatInd \\* pTransferStatInd](#)
- [BYTE \\* pReportDormStatus](#)
- [BYTE \\* pReportDataBearerTech](#)
- [BYTE \\* pReportChannelRate](#)

### 8.196.1 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>

## 8.196.2 Field Documentation

8.196.2.1 **ULONG** getDUNCallInfoReq::Mask8.196.2.2 **BYTE\*** getDUNCallInfoReq::pReportChannelRate8.196.2.3 **BYTE\*** getDUNCallInfoReq::pReportConnStatus8.196.2.4 **BYTE\*** getDUNCallInfoReq::pReportDataBearerTech8.196.2.5 **BYTE\*** getDUNCallInfoReq::pReportDormStatus8.196.2.6 **TransferStatInd\*** getDUNCallInfoReq::pTransferStatInd

## 8.197 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](#)

### 8.197.1 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>

<p><i>pLastCallData- BearerTech</i></p>	<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>
---	---

## 8.197.2 Field Documentation

8.197.2.1 **WORD\*** getDUNCallInfoResp::pCallEndReason

8.197.2.2 **ChannelRate\*** getDUNCallInfoResp::pChannelRate

8.197.2.3 **ConnectionStatus\*** getDUNCallInfoResp::pConnectionStatus

8.197.2.4 **BYTE\*** getDUNCallInfoResp::pDataBearerTech

8.197.2.5 **BYTE\*** getDUNCallInfoResp::pDormancyStatus

8.197.2.6 **BYTE\*** getDUNCallInfoResp::pLastCallDataBearerTech

8.197.2.7 **ULONGLONG\*** getDUNCallInfoResp::pLastCallRXOKBytesCnt

8.197.2.8 **ULONGLONG\*** getDUNCallInfoResp::pLastCallTXOKBytesCnt

8.197.2.9 **ULONGLONG\*** getDUNCallInfoResp::pMdmCallDurationActive

8.197.2.10 **ULONGLONG\*** getDUNCallInfoResp::pRXOKBytesCount

8.197.2.11 **ULONGLONG\*** getDUNCallInfoResp::pTXOKBytesCount

## 8.198 getDyingGaspCfg Struct Reference

## Data Fields

- [BYTE](#) \* [pDestSMSNum](#)
- [BYTE](#) \* [pDestSMSContent](#)

## 8.198.1 Detailed Description

This struture contains the TLV required to get the Dying GASP Config.

## Parameters

<i>OUT]</i>	<p>pDestSMSNum[OUT]</p> <ul style="list-style-type: none"> <li>• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>• Optional parameter.</li> </ul>
<i>OUT]</i>	<p>pDestSMSContent[OUT]</p> <ul style="list-style-type: none"> <li>• SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>• Optional parameter.</li> </ul>

## 8.198.2 Field Documentation

8.198.2.1 [BYTE](#)\* [getDyingGaspCfg::pDestSMSContent](#)

8.198.2.2 [BYTE](#)\* [getDyingGaspCfg::pDestSMSNum](#)

## 8.199 getDyingGaspStatistics Struct Reference

## Data Fields

- [ULONG](#) \* [pTimeStamp](#)
- [BYTE](#) \* [pSMSAttemptedFlag](#)

## 8.199.1 Detailed Description

This struture contains the TLV required to get the Dying GASP Statistics.

## Parameters

<i>OUT]</i>	<p>pDestSMSNum[OUT]</p> <ul style="list-style-type: none"> <li>• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>• Optional parameter.</li> </ul>
<i>OUT]</i>	<p>pDestSMSContent[OUT]</p> <ul style="list-style-type: none"> <li>• SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>• Optional parameter.</li> </ul>

## 8.199.2 Field Documentation

8.199.2.1 [BYTE](#)\* [getDyingGaspStatistics::pSMSAttemptedFlag](#)

8.199.2.2 **ULONG\*** `getDyingGaspStatistics::pTimeStamp`

## 8.200 GetErrRateResp Struct Reference

### Data Fields

- **WORD\*** `pCDMAFrameErrRate`
- **WORD\*** `pHDRPackErrRate`
- **BYTE\*** `pGSMBER`
- **BYTE\*** `pWCDMABER`

### 8.200.1 Detailed Description

This structure contains information about the SLQSGetErrorRate response parameters.

#### Parameters

<i>pCDMAFrameErrRate[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>

### 8.200.2 Field Documentation

8.200.2.1 **WORD\*** `GetErrRateResp::pCDMAFrameErrRate`

8.200.2.2 **BYTE\*** `GetErrRateResp::pGSMBER`

8.200.2.3 **WORD\*** `GetErrRateResp::pHDRPackErrRate`

8.200.2.4 **BYTE\*** `GetErrRateResp::pWCDMABER`

## 8.201 GetHRPDStatsResp Struct Reference

## Data Fields

- [DRCParams](#) \* [pDRCParams](#)
- [BYTE](#) \* [pUATI](#)
- [PilotSetData](#) \* [pPilotSetData](#)

### 8.201.1 Detailed Description

This structure contains information about the SLQSSwiGetHRPDStats response parameters.

#### Parameters

<i>pDRCParams</i> [Out]	<ul style="list-style-type: none"> <li>• See <a href="#">DRCParams</a> for more information.</li> </ul>
<i>pUATI</i> [Out]	<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</i> [Out]	<ul style="list-style-type: none"> <li>• See <a href="#">PilotSetData</a> for more information.</li> </ul>

### 8.201.2 Field Documentation

8.201.2.1 [DRCParams](#)\* [GetHRPDStatsResp::pDRCParams](#)

8.201.2.2 [PilotSetData](#)\* [GetHRPDStatsResp::pPilotSetData](#)

8.201.2.3 [BYTE](#)\* [GetHRPDStatsResp::pUATI](#)

## 8.202 GetIMSSMSConfigParams Struct Reference

## Data Fields

- [BYTE](#) \* [pSettingResp](#)
- [BYTE](#) \* [pSMSFormat](#)
- [BYTE](#) \* [pSMSOverIPNwInd](#)
- [BYTE](#) \* [pPhoneCtxtURILen](#)
- [BYTE](#) \* [pPhoneCtxtURI](#)

### 8.202.1 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>pSMSOverIPNW-Ind</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[IN/OUT]</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>

## 8.202.2 Field Documentation

8.202.2.1 **BYTE\*** GetIMSSMSConfigParams::pPhoneCtxtURI8.202.2.2 **BYTE\*** GetIMSSMSConfigParams::pPhoneCtxtURLen8.202.2.3 **BYTE\*** GetIMSSMSConfigParams::pSettingResp8.202.2.4 **BYTE\*** GetIMSSMSConfigParams::pSMSFormat8.202.2.5 **BYTE\*** GetIMSSMSConfigParams::pSMSOverIPNWInd

## 8.203 GetIMSUserConfigParams Struct Reference

## Data Fields

- BYTE \*** [pSettingResp](#)
- BYTE \*** [pIMSDomainLen](#)
- BYTE \*** [pIMSDomain](#)

## 8.203.1 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[IN/OUT]</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>
-------------------	---

### 8.203.2 Field Documentation

8.203.2.1 **BYTE\*** GetIMSUserConfigParams::pIMSDomain

8.203.2.2 **BYTE\*** GetIMSUserConfigParams::pIMSDomainLen

8.203.2.3 **BYTE\*** GetIMSUserConfigParams::pSettingResp

## 8.204 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

### 8.204.1 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>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>

## 8.204.2 Field Documentation

- 8.204.2.1 **BYTE\*** GetIMSVoIPConfigResp::pAmrMode
- 8.204.2.2 **BYTE\*** GetIMSVoIPConfigResp::pAmrOctetAligned
- 8.204.2.3 **BYTE\*** GetIMSVoIPConfigResp::pAmrWbEnable
- 8.204.2.4 **WORD\*** GetIMSVoIPConfigResp::pAmrWBMode
- 8.204.2.5 **BYTE\*** GetIMSVoIPConfigResp::pAmrWBOctetAligned
- 8.204.2.6 **WORD\*** GetIMSVoIPConfigResp::pMinSessionExpiryTimer
- 8.204.2.7 **WORD\*** GetIMSVoIPConfigResp::pRingBackTimer
- 8.204.2.8 **WORD\*** GetIMSVoIPConfigResp::pRingingTimer
- 8.204.2.9 **WORD\*** GetIMSVoIPConfigResp::pRTPRTCPInactTimer
- 8.204.2.10 **BYTE\*** GetIMSVoIPConfigResp::pScrAmrEnable
- 8.204.2.11 **BYTE\*** GetIMSVoIPConfigResp::pScrAmrWbEnable
- 8.204.2.12 **WORD\*** GetIMSVoIPConfigResp::pSessionExpiryTimer
- 8.204.2.13 **BYTE\*** GetIMSVoIPConfigResp::pSettingResp

## 8.205 GetInstIDResp Struct Reference

### Data Fields

- BYTE \*** [pInstanceId](#)
- BYTE \*** [pIPFamily](#)

## 8.205.1 Field Documentation

- 8.205.1.1 **BYTE\*** GetInstIDResp::pInstanceId

8.205.1.2 BYTE\* GetInstIDResp::pIPFamily

## 8.206 GetM2MAudioProfileReq Struct Reference

### Data Fields

- [BYTE \\* pGenerator](#)

### 8.206.1 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>
------------------------------	--

### 8.206.2 Field Documentation

8.206.2.1 BYTE\* GetM2MAudioProfileReq::pGenerator

## 8.207 GetM2MAudioProfileResp Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE Generator](#)
- [BYTE Volume](#)
- [BYTE CwtMute](#)

### 8.207.1 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>

## 8.207.2 Field Documentation

8.207.2.1 **BYTE** GetM2MAudioProfileResp::CwtMute

8.207.2.2 **BYTE** GetM2MAudioProfileResp::EarMute

8.207.2.3 **BYTE** GetM2MAudioProfileResp::Generator

8.207.2.4 **BYTE** GetM2MAudioProfileResp::MicMute

8.207.2.5 **BYTE** GetM2MAudioProfileResp::Profile

8.207.2.6 **BYTE** GetM2MAudioProfileResp::Volume

## 8.208 GetM2MAudioVolumeReq Struct Reference

### Data Fields

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

### 8.208.1 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>
------------------	--

### 8.208.2 Field Documentation

8.208.2.1 BYTE GetM2MAudioVolumeReq::Generator

8.208.2.2 BYTE GetM2MAudioVolumeReq::Profile

## 8.209 GetM2MAudioVolumeResp Struct Reference

### Data Fields

- [BYTE Level](#)

### 8.209.1 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>
--------------	--

### 8.209.2 Field Documentation

8.209.2.1 BYTE GetM2MAudioVolumeResp::Level

## 8.210 GetM2MAVMuteReq Struct Reference

### Data Fields

- [BYTE Profile](#)

### 8.210.1 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>
----------------	---

### 8.210.2 Field Documentation

8.210.2.1 **BYTE** GetM2MAVMuteReq::Profile

## 8.211 GetM2MAVMuteResp Struct Reference

### Data Fields

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

### 8.211.1 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>

### 8.211.2 Field Documentation

8.211.2.1 **BYTE** GetM2MAVMuteResp::CwtMute

8.211.2.2 **BYTE** GetM2MAVMuteResp::EarMute

8.211.2.3 **BYTE** GetM2MAVMuteResp::MicMute

## 8.212 GetM2MSpkrGainReq Struct Reference

### Data Fields

- [BYTE Profile](#)

### 8.212.1 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>
-----------------	---

## 8.212.2 Field Documentation

## 8.212.2.1 BYTE GetM2MSpkrGainReq::Profile

## 8.213 GetM2MSpkrGainResp Struct Reference

## Data Fields

- [WORD Value](#)

## 8.213.1 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>
--------------	--

## 8.213.2 Field Documentation

## 8.213.2.1 WORD GetM2MSpkrGainResp::Value

## 8.214 getMsgWaitingInfo Struct Reference

## Data Fields

- [BYTE numInstances](#)
- [messageWaitingInfoContent msgWaitInfo](#) [0xFF]

## 8.214.1 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>

### 8.214.2 Field Documentation

8.214.2.1 `messageWaitingInfoContent` `getMsgWaitingInfo::msgWaitInfo[0xFF]`

8.214.2.2 `BYTE` `getMsgWaitingInfo::numInstances`

## 8.215 GetNetworkTimeResp Struct Reference

### Data Fields

- [timeInfo](#) \* [p3GPP2TimeInfo](#)
- [timeInfo](#) \* [p3GPPTimeInfo](#)

### 8.215.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

#### Parameters

<i>p3GPP2TimeInfo</i>	[Optional] • See <a href="#">timeInfo</a> for more information
<i>p3GPPTimeInfo</i>	[Optional] • See <a href="#">timeInfo</a> for more information

### 8.215.2 Field Documentation

8.215.2.1 `timeInfo`\* `GetNetworkTimeResp::p3GPP2TimeInfo`

8.215.2.2 `timeInfo`\* `GetNetworkTimeResp::p3GPPTimeInfo`

## 8.216 GetRegMgrConfigParams Struct Reference

### Data Fields

- `BYTE` \* [pSettingResp](#)
- `WORD` \* [pPCSCFPort](#)
- `BYTE` \* [pPriCSCFPortNameLen](#)
- `BYTE` \* [pPriCSCFPortName](#)
- `BYTE` \* [pIMSTestMode](#)

### 8.216.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig response parameters.

#### Parameters

<i>pSettingResp</i>	• Settings Response
<i>pPCSCFPort</i>	• Proxy call session control function port

<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.

**8.216.2 Field Documentation**

8.216.2.1 **BYTE\*** GetRegMgrConfigParams::pIMSTestMode

8.216.2.2 **WORD\*** GetRegMgrConfigParams::pPCSCFPort

8.216.2.3 **BYTE\*** GetRegMgrConfigParams::pPriCSCFPortName

8.216.2.4 **BYTE\*** GetRegMgrConfigParams::pPriCSCFPortNameLen

8.216.2.5 **BYTE\*** GetRegMgrConfigParams::pSettingResp

**8.217 GetSessionIDResp Struct Reference****Data Fields**

- [ULONG](#) \* pSessionIDv4
- [ULONG](#) \* pSessionIDv6

**8.217.1 Field Documentation**

8.217.1.1 **ULONG\*** GetSessionIDResp::pSessionIDv4

8.217.1.2 **ULONG\*** GetSessionIDResp::pSessionIDv6

**8.218 GetSIPConfigResp Struct Reference****Data Fields**

- [BYTE](#) \* pSettingResp
- [WORD](#) \* pSIPLocalPort
- [ULONG](#) \* pTimerSIPReg
- [ULONG](#) \* pSubscribeTimer
- [ULONG](#) \* pTimerT1

- [ULONG \\* pTimerT2](#)
- [ULONG \\* pTimerTf](#)
- [BYTE \\* pSigCompEnabled](#)

### 8.218.1 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>

### 8.218.2 Field Documentation

8.218.2.1 **BYTE\*** GetSIPConfigResp::pSettingResp

8.218.2.2 **BYTE\*** GetSIPConfigResp::pSigCompEnabled

8.218.2.3 **WORD\*** GetSIPConfigResp::pSIPLocalPort

8.218.2.4 **ULONG\*** GetSIPConfigResp::pSubscribeTimer

8.218.2.5 **ULONG\*** GetSIPConfigResp::pTimerSIPReg

8.218.2.6 **ULONG\*** GetSIPConfigResp::pTimerT1

8.218.2.7 **ULONG\*** GetSIPConfigResp::pTimerT2

8.218.2.8 **ULONG\*** GetSIPConfigResp::pTimerTf

## 8.219 GnssData Struct Reference

### Data Fields

- [ULONGLONG mask](#)

### 8.219.1 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 GLONASS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLONASS 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>
-------------	---

## 8.219.2 Field Documentation

### 8.219.2.1 ULONGLONG GnssData::mask

## 8.220 gnssSvInfoNotification Struct Reference

## Data Fields

- [BYTE bAltitudeAssumed](#)
- [satelliteInfo](#) \* [pSatelliteInfo](#)

### 8.220.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

#### Parameters

<i>bAltitude-Assumed</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

### 8.220.2 Field Documentation

8.220.2.1 **BYTE** gnssSvInfoNotification::bAltitudeAssumed

8.220.2.2 **satelliteInfo**\* gnssSvInfoNotification::pSatelliteInfo

## 8.221 GPRSQoS Struct Reference

## Data Fields

- [ULONG precedenceClass](#)
- [ULONG delayClass](#)
- [ULONG reliabilityClass](#)
- [ULONG peakThroughputClass](#)
- [ULONG meanThroughputClass](#)

### 8.221.1 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>

## 8.221.2 Field Documentation

8.221.2.1 **ULONG** GPRSQoS::delayClass

8.221.2.2 **ULONG** GPRSQoS::meanThroughputClass

8.221.2.3 **ULONG** GPRSQoS::peakThroughputClass

8.221.2.4 **ULONG** GPRSQoS::precedenceClass

8.221.2.5 **ULONG** GPRSQoS::reliabilityClass

## 8.222 GPRSRequestedQoS Struct Reference

### Data Fields

- [ULONG precedenceClass](#)
- [ULONG delayClass](#)
- [ULONG reliabilityClass](#)
- [ULONG peakThroughputClass](#)
- [ULONG meanThroughputClass](#)

## 8.222.1 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>
------------------------------	---

### 8.222.2 Field Documentation

8.222.2.1 **ULONG** GPRSRequestedQoS::delayClass

8.222.2.2 **ULONG** GPRSRequestedQoS::meanThroughputClass

8.222.2.3 **ULONG** GPRSRequestedQoS::peakThroughputClass

8.222.2.4 **ULONG** GPRSRequestedQoS::precedenceClass

8.222.2.5 **ULONG** GPRSRequestedQoS::reliabilityClass

## 8.223 GPSStateInfo 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

### 8.223.1 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>lono_valid</i>	<ul style="list-style-type: none"> <li>• lono 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 <a href="#">SV</a> mask for ephemeris; if the bit is set, ephemeris for that <a href="#">SV</a> is available.</li> </ul>
<i>gps_almanac_ - sv_msk</i>	<ul style="list-style-type: none"> <li>• GPS <a href="#">SV</a> mask for almanac; if the bit is set, almanac for that <a href="#">SV</a> is available.</li> </ul>
<i>gps_health_sv_ - msk</i>	<ul style="list-style-type: none"> <li>• GPS <a href="#">SV</a> mask for health; if the bit is set, health for that <a href="#">SV</a> is available.</li> </ul>
<i>gps_visible_sv_ - msk</i>	<ul style="list-style-type: none"> <li>• GPS <a href="#">SV</a> mask for visible Svs; if the bit is set, the <a href="#">SV</a> is available.</li> </ul>
<i>glo_ephemeris_ - sv_msk</i>	<ul style="list-style-type: none"> <li>• GLONASS <a href="#">SV</a> mask for ephemeris; if the bit is set, ephemeris for that <a href="#">SV</a> is available.</li> </ul>
<i>glo_almanac_sv_ - _msk</i>	<ul style="list-style-type: none"> <li>• GLONASS <a href="#">SV</a> mask for almanac; if the bit is set, almanac for that <a href="#">SV</a> is available.</li> </ul>
<i>glo_health_sv_ - msk</i>	<ul style="list-style-type: none"> <li>• GLONASS <a href="#">SV</a> mask for health; if the bit is set, health for that <a href="#">SV</a> is available.</li> </ul>
<i>glo_visible_sv_ - msk</i>	<ul style="list-style-type: none"> <li>• GLONASS <a href="#">SV</a> mask for visible SVs; if the bit is set, the <a href="#">SV</a> is available.</li> </ul>
<i>sbas_ - ephemeris_sv_ - msk</i>	<ul style="list-style-type: none"> <li>• SBAS <a href="#">SV</a> mask for ephemeris; if the bit is set, ephemeris for that <a href="#">SV</a> is available.</li> </ul>

<i>sbas_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>• SBAS <a href="#">SV</a> mask for almanac; if the bit is set, almanac for that <a href="#">SV</a> is available.</li> </ul>
<i>sbas_health_sv_msk</i>	<ul style="list-style-type: none"> <li>• SBAS <a href="#">SV</a> mask for health; if the bit is set, health for that <a href="#">SV</a> is available.</li> </ul>
<i>sbas_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>• SBAS <a href="#">SV</a> mask for visible SVs; if the bit is set, the <a href="#">SV</a> 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>

## 8.223.2 Field Documentation

8.223.2.1 **ULONG** GPSSStateInfo::Altitude

8.223.2.2 **BYTE** GPSSStateInfo::EngineState

8.223.2.3 **ULONG** GPSSStateInfo::glo\_almanac\_sv\_msk

8.223.2.4 **ULONG** GPSSStateInfo::glo\_ephemeris\_sv\_msk

8.223.2.5 **ULONG** GPSSStateInfo::glo\_health\_sv\_msk

8.223.2.6 **ULONG** GPSSStateInfo::glo\_visible\_sv\_msk

8.223.2.7 **ULONG** GPSSStateInfo::gps\_almanac\_sv\_msk

8.223.2.8 **ULONG** GPSSStateInfo::gps\_ephemeris\_sv\_msk

8.223.2.9 **ULONG** GPSSStateInfo::gps\_health\_sv\_msk

8.223.2.10 **ULONG** GPSSStateInfo::gps\_visible\_sv\_msk

8.223.2.11 **ULONG** GPSSStateInfo::HorizontalUncertainty

8.223.2.12 **BYTE** GPSSStateInfo::lono\_valid

8.223.2.13 **ULONGLONG** GPSSStateInfo::Latitude

8.223.2.14 **ULONGLONG** GPSSStateInfo::Longitude

8.223.2.15 **ULONG** GPSSStateInfo::sbas\_almanac\_sv\_msk

8.223.2.16 **ULONG** GPSSStateInfo::sbas\_ephemeris\_sv\_msk

- 8.223.2.17 **ULONG** GPSSStateInfo::sbas\_health\_sv\_msk
- 8.223.2.18 **ULONG** GPSSStateInfo::sbas\_visible\_sv\_msk
- 8.223.2.19 **ULONG** GPSSStateInfo::Time\_uncert\_ms
- 8.223.2.20 **WORD** GPSSStateInfo::TimeStmp\_gps\_week
- 8.223.2.21 **ULONG** GPSSStateInfo::TimeStmp\_tow\_ms
- 8.223.2.22 **ULONG** GPSSStateInfo::ValidMask
- 8.223.2.23 **ULONG** GPSSStateInfo::VerticalUncertainty
- 8.223.2.24 **WORD** GPSSStateInfo::xtra\_start\_gps\_minutes
- 8.223.2.25 **WORD** GPSSStateInfo::xtra\_start\_gps\_week
- 8.223.2.26 **WORD** GPSSStateInfo::xtra\_valid\_duration\_hours

## 8.224 gpsTime\_s Struct Reference

### Data Fields

- [WORD](#) gpsWeek
- [ULONG](#) gpsTimeOfWeekMs

### 8.224.1 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>

### 8.224.2 Field Documentation

- 8.224.2.1 **ULONG** gpsTime\_s::gpsTimeOfWeekMs
- 8.224.2.2 **WORD** gpsTime\_s::gpsWeek

## 8.225 gsmCellInfo Struct Reference

### Data Fields

- [WORD](#) arfcn
- [BYTE](#) band1900

- [BYTE cellIdValid](#)
- [BYTE bsicId](#)
- [SHORT rssi](#)
- [SHORT srxlev](#)

### 8.225.1 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 arfcn 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>

### 8.225.2 Field Documentation

8.225.2.1 **WORD** gsmCellInfo::arfcn

8.225.2.2 **BYTE** gsmCellInfo::band1900

8.225.2.3 **BYTE** gsmCellInfo::bsicId

8.225.2.4 **BYTE** gsmCellInfo::cellIdValid

8.225.2.5 **SHORT** gsmCellInfo::rssi

8.225.2.6 **SHORT** gsmCellInfo::srxlev

## 8.226 GSMRSSIThresh Struct Reference

### Data Fields

- [BYTE GSMRSSIThreshListLen](#)
- [WORD \\* pGSMRSSIThreshList](#)

### 8.226.1 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>

### 8.226.2 Field Documentation

8.226.2.1 **BYTE** GSMRSSIThresh::GSMRSSIThreshListLen

8.226.2.2 **WORD\*** GSMRSSIThresh::pGSMRSSIThreshList

## 8.227 GSMSrvStatusInfo Struct Reference

### Data Fields

- [BYTE srvStatus](#)
- [BYTE trueSrvStatus](#)
- [BYTE isPrefDataPath](#)

### 8.227.1 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>

## 8.227.2 Field Documentation

8.227.2.1 **BYTE** GSMSrvStatusInfo::isPrefDataPath

8.227.2.2 **BYTE** GSMSrvStatusInfo::srvStatus

8.227.2.3 **BYTE** GSMSrvStatusInfo::trueSrvStatus

## 8.228 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

### 8.228.1 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>
<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> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 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>

## 8.228.2 Field Documentation

8.228.2.1 **ULONG** GSMSysInfo::cellId

8.228.2.2 **BYTE** GSMSysInfo::cellIdValid

8.228.2.3 **BYTE** GSMSysInfo::dtmSupp

8.228.2.4 **BYTE** GSMSysInfo::dtmSuppValid

8.228.2.5 **BYTE** GSMSysInfo::egprsSupp

8.228.2.6 **BYTE** GSMSysInfo::egprsSuppValid

8.228.2.7 WORD GSMSysInfo::lac

8.228.2.8 BYTE GSMSysInfo::lacValid

8.228.2.9 BYTE GSMSysInfo::MCC[3]

8.228.2.10 BYTE GSMSysInfo::MNC[3]

8.228.2.11 BYTE GSMSysInfo::networkIdValid

8.228.2.12 BYTE GSMSysInfo::regRejectInfoValid

8.228.2.13 BYTE GSMSysInfo::rejCause

8.228.2.14 BYTE GSMSysInfo::rejectSrvDomain

8.228.2.15 sysInfoCommon GSMSysInfo::sysInfoGSM

## 8.229 gyroAcceptReady\_s Struct Reference

### Data Fields

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

### 8.229.1 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 <ul style="list-style-type: none"> <li>• 0x01 - Ready to accept sensor data</li> <li>• 0x00 - Not ready to accept sensor data</li> </ul> </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>

### 8.229.2 Field Documentation

8.229.2.1 WORD gyroAcceptReady\_s::batchPerSec

8.229.2.2 **BYTE** gyroAcceptReady\_s::injectEnable

8.229.2.3 **WORD** gyroAcceptReady\_s::samplesPerBatch

## 8.230 gyroTempAcceptReady\_s Struct Reference

### Data Fields

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

### 8.230.1 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 <ul style="list-style-type: none"> <li>• 0x01 - Ready to accept sensor data</li> <li>• 0x00 - Not ready to accept sensor data</li> </ul> </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>

### 8.230.2 Field Documentation

8.230.2.1 **WORD** gyroTempAcceptReady\_s::batchPerSec

8.230.2.2 **BYTE** gyroTempAcceptReady\_s::injectEnable

8.230.2.3 **WORD** gyroTempAcceptReady\_s::samplesPerBatch

## 8.231 HDRECIOTresh Struct Reference

### Data Fields

- [BYTE HDRECIOTreshListLen](#)
- [WORD \\* pHDRICIOTreshList](#)

### 8.231.1 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>

### 8.231.2 Field Documentation

8.231.2.1 **BYTE** HDRECIOThresh::HDRECIOThreshListLen

8.231.2.2 **WORD\*** HDRECIOThresh::pHDRECIOThreshList

## 8.232 HDRIOThresh Struct Reference

#### Data Fields

- [BYTE HDRIOThreshListLen](#)
- [WORD \\* pHDRIOThreshList](#)

### 8.232.1 Detailed Description

This structure contains HDR IO threshold related parameters.

#### Parameters

<i>HDRIOThresh- ListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR IO threshold list parameter to follow</li> </ul>
<i>pHDRIOThresh- 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>

### 8.232.2 Field Documentation

8.232.2.1 **BYTE** HDRIOThresh::HDRIOThreshListLen

8.232.2.2 **WORD\*** HDRIOThresh::pHDRIOThreshList

## 8.233 HDRPersonalityInd Struct Reference

## Data Fields

- [WORD](#) \* [pCurrentPersonality](#)
- [BYTE](#) \* [pPersonalityListLength](#)
- [protocolSubtypeElement](#) \* [pProtocolSubtypeElement](#)

## 8.233.1 Field Documentation

8.233.1.1 [WORD](#)\* HDRPersonalityInd::pCurrentPersonality8.233.1.2 [BYTE](#)\* HDRPersonalityInd::pPersonalityListLength8.233.1.3 [protocolSubtypeElement](#)\* HDRPersonalityInd::pProtocolSubtypeElement

## 8.234 HDRPersonalityResp Struct Reference

## Data Fields

- [WORD](#) \* [pCurrentPersonality](#)
- [BYTE](#) \* [pPersonalityListLength](#)
- [protocolSubtypeElement](#) \* [pProtocolSubtypeElement](#)

## 8.234.1 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>

## 8.234.2 Field Documentation

8.234.2.1 [WORD](#)\* HDRPersonalityResp::pCurrentPersonality8.234.2.2 [BYTE](#)\* HDRPersonalityResp::pPersonalityListLength8.234.2.3 [protocolSubtypeElement](#)\* HDRPersonalityResp::pProtocolSubtypeElement

## 8.235 HDRProtSubtypResp Struct Reference

## Data Fields

- [WORD](#) \* [pCurrentPrsnlty](#)
- [BYTE](#) \* [pPersonalityListLength](#)

- [protocolSubtypeElement](#) \* [pProtoSubTypElmnt](#)
- [ULONGLONG](#) \* [pAppSubType](#)

### 8.235.1 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>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 4</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>
<i>pAppSubType[-Out]</i>	<ul style="list-style-type: none"> <li>• Stream application subtype</li> <li>• Application subtype for each stream,</li> </ul>

### 8.235.2 Field Documentation

8.235.2.1 [ULONGLONG](#)\* HDRProtSubtypResp::pAppSubType

8.235.2.2 [WORD](#)\* HDRProtSubtypResp::pCurrentPrsnlty

8.235.2.3 [BYTE](#)\* HDRProtSubtypResp::pPersonalityListLength

8.235.2.4 [protocolSubtypeElement](#)\* HDRProtSubtypResp::pProtoSubTypElmnt

## 8.236 HDRRSSIThresh Struct Reference

#### Data Fields

- [BYTE](#) HDRRSSIThreshListLen
- [WORD](#) \* [pHDRRSSIThreshList](#)

### 8.236.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

## Parameters

<i>HDRRSSI- ThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR RSSI threshold list parameter to follow</li> </ul>
<i>pHDRRSSI- 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: -118 to -13 (in dBm).</li> </ul>

## 8.236.2 Field Documentation

8.236.2.1 BYTE HDRRSSIthresh::HDRRSSIthreshListLen

8.236.2.2 WORD\* HDRRSSIthresh::pHDRRSSIthreshList

## 8.237 HDRSINRThresh Struct Reference

## Data Fields

- [BYTE HDRSINRThresListLen](#)
- [BYTE \\* pHDRSINRThresList](#)

## 8.237.1 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>

## 8.237.2 Field Documentation

8.237.2.1 BYTE HDRSINRthresh::HDRSINRThresListLen

8.237.2.2 BYTE\* HDRSINRthresh::pHDRSINRThresList

## 8.238 HDRSINRThreshold Struct Reference

## Data Fields

- [BYTE HDRSINRThreshListLen](#)
- [WORD \\* pHDRSINRThreshList](#)

### 8.238.1 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>

### 8.238.2 Field Documentation

8.238.2.1 **BYTE** HDRSINRThreshold::HDRSINRThreshListLen

8.238.2.2 **WORD\*** HDRSINRThreshold::pHDRSINRThreshList

## 8.239 HDRSSInfo Struct Reference

## Data Fields

- [INT8 rssi](#)
- [SHORT ecio](#)
- [BYTE sinr](#)
- [INT32 io](#)

### 8.239.1 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>

## 8.239.2 Field Documentation

## 8.239.2.1 SHORT HDRSSInfo::ecio

## 8.239.2.2 INT32 HDRSSInfo::io

## 8.239.2.3 INT8 HDRSSInfo::rssI

## 8.239.2.4 BYTE HDRSSInfo::sinr

## 8.240 hdrSSInfo Struct Reference

## Data Fields

- int8\_t [rssI](#)
- int16\_t [ecio](#)
- uint8\_t [sinr](#)
- int32\_t [io](#)

## 8.240.1 Detailed Description

## Parameters

<i>rssI</i>	RSSI in dBm.
<i>ecio</i>	ECIO value representing negative 0.5 dBm increment
<i>sinr</i>	SINR level.
<i>io</i>	Received IO in dBm.

## 8.240.2 Field Documentation

8.240.2.1 int16\_t hdrSSInfo::ecio

8.240.2.2 int32\_t hdrSSInfo::io

8.240.2.3 int8\_t hdrSSInfo::rssI

8.240.2.4 uint8\_t hdrSSInfo::sinr

## 8.241 HDRSysInfo Struct Reference

## Data Fields

- [sysInfoCommon sysInfoHDR](#)
- [BYTE isSysPrIMatchValid](#)
- [BYTE isSysPrIMatch](#)
- [BYTE hdrPersonalityValid](#)
- [BYTE hdrPersonality](#)
- [BYTE hdrActiveProtValid](#)
- [BYTE hdrActiveProt](#)
- [BYTE is856SysIdValid](#)
- [BYTE is856SysId](#) [16]

## 8.241.1 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>

### 8.241.2 Field Documentation

- 8.241.2.1 **BYTE** HDRSysInfo::hdrActiveProt
- 8.241.2.2 **BYTE** HDRSysInfo::hdrActiveProtValid
- 8.241.2.3 **BYTE** HDRSysInfo::hdrPersonality
- 8.241.2.4 **BYTE** HDRSysInfo::hdrPersonalityValid
- 8.241.2.5 **BYTE** HDRSysInfo::is856SysId[16]
- 8.241.2.6 **BYTE** HDRSysInfo::is856SysIdValid
- 8.241.2.7 **BYTE** HDRSysInfo::isSysPrIMatch
- 8.241.2.8 **BYTE** HDRSysInfo::isSysPrIMatchValid
- 8.241.2.9 **sysInfoCommon** HDRSysInfo::sysInfoHDR

## 8.242 homeSIDNID Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [sidNid](#) SidNid [255]

### 8.242.1 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>

### 8.242.2 Field Documentation

- 8.242.2.1 **BYTE** homeSIDNID::numInstances
- 8.242.2.2 **sidNid** homeSIDNID::SidNid[255]

## 8.243 hotSwapStatus Struct Reference

## Data Fields

- [BYTE hotSwapLength](#)
- [BYTE hotSwap](#) [255]

### 8.243.1 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>

### 8.243.2 Field Documentation

8.243.2.1 [BYTE hotSwapStatus::hotSwap](#)[255]

8.243.2.2 [BYTE hotSwapStatus::hotSwapLength](#)

## 8.244 image\_info\_t Struct Reference

## Data Fields

- [uint8\\_t imageType](#)
- [uint8\\_t uniqueID](#) [16]
- [uint8\\_t buildIDLen](#)
- [uint8\\_t buildID](#) [255]

### 8.244.1 Field Documentation

8.244.1.1 [uint8\\_t image\\_info\\_t::buildID](#)[255]

8.244.1.2 [uint8\\_t image\\_info\\_t::buildIDLen](#)

8.244.1.3 [uint8\\_t image\\_info\\_t::imageType](#)

8.244.1.4 [uint8\\_t image\\_info\\_t::uniqueID](#)[16]

## 8.245 ImageElement Struct Reference

## Data Fields

- [BYTE imageType](#)

- [BYTE imageId](#) [16]
- [BYTE buildIdLength](#)
- [CHAR buildId](#) [100]

### 8.245.1 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>buildId</i>	<ul style="list-style-type: none"> <li>• Build ID ANSI string( Max 100 characters )</li> </ul>

### 8.245.2 Field Documentation

8.245.2.1 [CHAR ImageElement::buildId](#)[100]

8.245.2.2 [BYTE ImageElement::buildIdLength](#)

8.245.2.3 [BYTE ImageElement::imageId](#)[16]

8.245.2.4 [BYTE ImageElement::imageType](#)

## 8.246 ImageIdElement Struct Reference

### Data Fields

- [BYTE storageIndex](#)
- [BYTE failureCount](#)
- [BYTE imageID](#) [16]
- [BYTE buildIDLength](#)
- [CHAR buildID](#) [100]

### 8.246.1 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>

## 8.246.2 Field Documentation

8.246.2.1 **CHAR** ImageIDElement::buildID[100]

8.246.2.2 **BYTE** ImageIDElement::buildIDLength

8.246.2.3 **BYTE** ImageIDElement::failureCount

8.246.2.4 **BYTE** ImageIDElement::imageID[16]

8.246.2.5 **BYTE** ImageIDElement::storageIndex

## 8.247 ImageIDEntries Struct Reference

### Data Fields

- [BYTE](#) *imageType*
- [BYTE](#) *maxImages*
- [BYTE](#) *executingImage*
- [BYTE](#) *imageIDSize*
- struct [ImageIDElement](#) *imageIDElement* [50]

### 8.247.1 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>

### 8.247.2 Field Documentation

8.247.2.1 **BYTE** ImageIDEntries::executingImage

8.247.2.2 **struct** ImageIDElement ImageIDEntries::imageIDElement[50]

8.247.2.3 **BYTE** ImageIDEntries::imageIDSize

8.247.2.4 **BYTE** ImageIDEntries::imageType

8.247.2.5 **BYTE** ImageIDEntries::maxImages

## 8.248 ImageList Struct Reference

### Data Fields

- [BYTE](#) *listSize*
- struct** [ImageIDEntries](#) *imageIDEntries* [2]

### 8.248.1 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>

### 8.248.2 Field Documentation

8.248.2.1 **struct** ImageIDEntries ImageList::imageIDEntries[2]

8.248.2.2 **BYTE** ImageList::listSize

## 8.249 IMSAIndRegisterInfo Struct Reference

### Data Fields

- BYTE** \* *pRegStatusConfig*
- BYTE** \* *pServiceStatusConfig*
- BYTE** \* *pRatHandoverStatusConfig*

- [BYTE \\* pPdpStatusConfig](#)

### 8.249.1 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.

### 8.249.2 Field Documentation

8.249.2.1 **BYTE\*** IMSAIndRegisterInfo::pPdpStatusConfig

8.249.2.2 **BYTE\*** IMSAIndRegisterInfo::pRatHandoverStatusConfig

8.249.2.3 **BYTE\*** IMSAIndRegisterInfo::pRegStatusConfig

8.249.2.4 **BYTE\*** IMSAIndRegisterInfo::pServiceStatusConfig

## 8.250 imsaPdpStatusInfo Struct Reference

#### Data Fields

- [BYTE connetionState](#)

- [ULONG \\* pFailErrorCode](#)

### 8.250.1 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>

### 8.250.2 Field Documentation

8.250.2.1 **BYTE** `imsaPdpStatusInfo::connetionState`

8.250.2.2 **ULONG\*** `imsaPdpStatusInfo::pFailErrorCode`

## 8.251 imsaRatStatusInfo Struct Reference

### Data Fields

- [ULONG \\* pRATStatus](#)
- [ULONG \\* pSrcRAT](#)
- [ULONG \\* pTgtRAT](#)
- [BYTE \\* pErrorCodeStr](#)

### 8.251.1 Detailed Description

Contains the parameters passed for SLQSSetIMSAratStatusCallback 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>
----------------------	---

## 8.251.2 Field Documentation

8.251.2.1 **BYTE\*** `imsaRatStatusInfo::pErrorCodeStr`

8.251.2.2 **ULONG\*** `imsaRatStatusInfo::pRATStatus`

8.251.2.3 **ULONG\*** `imsaRatStatusInfo::pSrcRAT`

8.251.2.4 **ULONG\*** `imsaRatStatusInfo::pTgtRAT`

## 8.252 IMSARegistrationStatus Struct Reference

### Data Fields

- BYTE** \* `plmsRegStatus`
- WORD** \* `plmsRegErrCode`
- ULONG** \* `pNewImsRegStatus`

### 8.252.1 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>plmsRegErrCode</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 <code>IMSA_STATUS_NOT_REGISTERED</code>. -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>pNewImsRegStatus</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>

## 8.252.2 Field Documentation

8.252.2.1 **WORD\*** IMSARegistrationStatus::plmsRegErrCode

8.252.2.2 **BYTE\*** IMSARegistrationStatus::plmsRegStatus

8.252.2.3 **ULONG\*** IMSARegistrationStatus::pNewlmsRegStatus

## 8.253 imsaRegStatusInfo Struct Reference

### Data Fields

- **BYTE \*** [pbIMSRegistered](#)
- **WORD \*** [pRegStatusErrorCode](#)
- **ULONG \*** [plmsRegStatus](#)

### 8.253.1 Detailed Description

Contains the parameters passed for SLQSSetIMSARegStatusCallback by the device.

#### Parameters

<i>pbIMSRegistered</i>	<ul style="list-style-type: none"> <li>• TRUE/FALSE</li> </ul>
<i>pRegStatusErrorCode</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>plmsRegStatus</i>	IMS registration status. Values: IMSA_STATUS_NOT_REGISTERED - 0 IMSA_STATUS_REGISTERING - 1 IMSA_STATUS_REGISTERED -2

## 8.253.2 Field Documentation

8.253.2.1 **BYTE\*** imsaRegStatusInfo::pbIMSRegistered

8.253.2.2 **ULONG\*** imsaRegStatusInfo::plmsRegStatus

8.253.2.3 **WORD\*** imsaRegStatusInfo::pRegStatusErrorCode

## 8.254 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](#)

## 8.254.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

## Parameters

<i>pSmsService-Status</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>pVoipService-Status</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>pVtService-Status</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>

## 8.254.2 Field Documentation

8.254.2.1 **ULONG\*** IMSAServiceStatus::pSmsServiceRat

8.254.2.2 **ULONG\*** IMSAServiceStatus::pSmsServiceStatus

8.254.2.3 **ULONG\*** IMSAServiceStatus::pUtServiceRat

8.254.2.4 **ULONG\*** IMSAServiceStatus::pUtServiceStatus

8.254.2.5 **ULONG\*** IMSAServiceStatus::pVoipServiceRat

8.254.2.6 **ULONG\*** IMSAServiceStatus::pVoipServiceStatus

8.254.2.7 **ULONG\*** IMSAServiceStatus::pVsServiceRat

8.254.2.8 **ULONG\*** IMSAServiceStatus::pVsServiceStatus

8.254.2.9 **ULONG\*** IMSAServiceStatus::pVtServiceRat

8.254.2.10 **ULONG\*** IMSAServiceStatus::pVtServiceStatus

## 8.255 IMSASupportedFieldsResp Struct Reference

### Data Fields

- struct [ReqFieldsList](#) \* pReqFieldsList
- struct [RespFieldsList](#) \* pRespFieldsList
- struct [IndFieldsList](#) \* pIndFieldsList

### 8.255.1 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>

### 8.255.2 Field Documentation

8.255.2.1 struct IndFieldsList\* IMSASupportedFieldsResp::pIndFieldsList

8.255.2.2 struct ReqFieldsList\* IMSASupportedFieldsResp::pReqFieldsList

8.255.2.3 struct RespFieldsList\* IMSASupportedFieldsResp::pRespFieldsList

## 8.256 IMSASupportedMsgInfo Struct Reference

#### Data Fields

- struct [SupportedMsgList](#) \* [pSupportedMsgList](#)

### 8.256.1 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>
--------------------------	--

### 8.256.2 Field Documentation

8.256.2.1 struct SupportedMsgList\* IMSASupportedMsgInfo::pSupportedMsgList

## 8.257 imsaSvcStatusInfo Struct Reference

#### Data Fields

- [ULONG](#) \* [pSMSSvcStatus](#)

- [ULONG \\* pVOIPSvcStatus](#)
- [ULONG \\* pVTSvcStatus](#)
- [ULONG \\* pSMSSvcRAT](#)
- [ULONG \\* pVOIPSvcRAT](#)
- [ULONG \\* pVTSvcRAT](#)
- [ULONG \\* pUTSvcStatus](#)
- [ULONG \\* pUTSvcRAT](#)

### 8.257.1 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

### 8.257.2 Field Documentation

8.257.2.1 **ULONG\*** [imsaSvcStatusInfo::pSMSSvcRAT](#)

8.257.2.2 **ULONG\*** [imsaSvcStatusInfo::pSMSSvcStatus](#)

8.257.2.3 **ULONG\*** [imsaSvcStatusInfo::pUTSvcRAT](#)

8.257.2.4 **ULONG\*** [imsaSvcStatusInfo::pUTSvcStatus](#)

8.257.2.5 **ULONG\*** [imsaSvcStatusInfo::pVOIPSvcRAT](#)

8.257.2.6 **ULONG\*** [imsaSvcStatusInfo::pVOIPSvcStatus](#)

8.257.2.7 **ULONG\*** [imsaSvcStatusInfo::pVTSvcRAT](#)

8.257.2.8 **ULONG\*** [imsaSvcStatusInfo::pVTSvcStatus](#)

## 8.258 imsCfgIndRegisterInfo Struct Reference

#### Data Fields

- [BYTE \\* pSIPConfigEvents](#)
- [BYTE \\* pRegMgrConfigEvents](#)
- [BYTE \\* pSMSConfigEvents](#)
- [BYTE \\* pUserConfigEvents](#)
- [BYTE \\* pVoIPConfigEvents](#)

### 8.258.1 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.

### 8.258.2 Field Documentation

**8.258.2.1** BYTE\* imsCfgIndRegisterInfo::pRegMgrConfigEvents

**8.258.2.2** BYTE\* imsCfgIndRegisterInfo::pSIPConfigEvents

**8.258.2.3** BYTE\* imsCfgIndRegisterInfo::pSMSConfigEvents

8.258.2.4 **BYTE\*** `imsCfgIndRegisterInfo::pUserConfigEvents`

8.258.2.5 **BYTE\*** `imsCfgIndRegisterInfo::pVoIPConfigEvents`

## 8.259 `imsRegMgrConfigInfo` Struct Reference

### Data Fields

- **WORD\*** `pPriCSCFPort`
- **BYTE\*** `pCSCFPortName`
- **BYTE\*** `pIMSTestMode`

### 8.259.1 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>

### 8.259.2 Field Documentation

8.259.2.1 **BYTE\*** `imsRegMgrConfigInfo::pCSCFPortName`

8.259.2.2 **BYTE\*** `imsRegMgrConfigInfo::pIMSTestMode`

8.259.2.3 **WORD\*** `imsRegMgrConfigInfo::pPriCSCFPort`

## 8.260 `imsSIPConfigInfo` Struct Reference

### Data Fields

- **WORD\*** `pSIPLocalPort`
- **ULONG\*** `pTimerSIPReg`
- **ULONG\*** `pSubscribeTimer`
- **ULONG\*** `pTimerT1`
- **ULONG\*** `pTimerT2`
- **ULONG\*** `pTimerTf`
- **BYTE\*** `pSigCompEnabled`

### 8.260.1 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>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>

#### Note

None

### 8.260.2 Field Documentation

8.260.2.1 **BYTE\*** imsSIPConfigInfo::pSigCompEnabled

8.260.2.2 **WORD\*** imsSIPConfigInfo::pSIPLocalPort

8.260.2.3 **ULONG\*** imsSIPConfigInfo::pSubscribeTimer

8.260.2.4 **ULONG\*** imsSIPConfigInfo::pTimerSIPReg

8.260.2.5 **ULONG\*** imsSIPConfigInfo::pTimerT1

8.260.2.6 **ULONG\*** imsSIPConfigInfo::pTimerT2

8.260.2.7 **ULONG\*** imsSIPConfigInfo::pTimerTf

## 8.261 imsSMSConfigInfo Struct Reference

#### Data Fields

- BYTE \*** pSMSFormat

- [BYTE \\* pSMSOverIPNwInd](#)
- [BYTE \\* pPhoneCtxtURI](#)

### 8.261.1 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>

### 8.261.2 Field Documentation

8.261.2.1 [BYTE\\*](#) imsSMSConfigInfo::pPhoneCtxtURI

8.261.2.2 [BYTE\\*](#) imsSMSConfigInfo::pSMSFormat

8.261.2.3 [BYTE\\*](#) imsSMSConfigInfo::pSMSOverIPNwInd

## 8.262 imsUserConfigInfo Struct Reference

### Data Fields

- [BYTE \\* pIMSDomain](#)

### 8.262.1 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>
-------------------	--

### 8.262.2 Field Documentation

8.262.2.1 [BYTE\\*](#) imsUserConfigInfo::pIMSDomain

## 8.263 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

### 8.263.1 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>

### 8.263.2 Field Documentation

- 8.263.2.1 **BYTE\*** `imsVoIPConfigInfo::pAmrMode`
- 8.263.2.2 **BYTE\*** `imsVoIPConfigInfo::pAmrOctetAligned`
- 8.263.2.3 **BYTE\*** `imsVoIPConfigInfo::pAmrWbEnable`
- 8.263.2.4 **WORD\*** `imsVoIPConfigInfo::pAmrWBMode`
- 8.263.2.5 **BYTE\*** `imsVoIPConfigInfo::pAmrWBOctetAligned`
- 8.263.2.6 **WORD\*** `imsVoIPConfigInfo::pMinSessionExpiryTimer`
- 8.263.2.7 **WORD\*** `imsVoIPConfigInfo::pRingBackTimer`
- 8.263.2.8 **WORD\*** `imsVoIPConfigInfo::pRingingTimer`
- 8.263.2.9 **WORD\*** `imsVoIPConfigInfo::pRTPRTCPInactTimer`
- 8.263.2.10 **BYTE\*** `imsVoIPConfigInfo::pScrAmrEnable`
- 8.263.2.11 **BYTE\*** `imsVoIPConfigInfo::pScrAmrWbEnable`
- 8.263.2.12 **WORD\*** `imsVoIPConfigInfo::pSessionExpiryTimer`

## 8.264 IndFieldsList Struct Reference

### Data Fields

- [BYTE indicationFieldsLen](#)
- [BYTE indicationFields](#) [256]

### 8.264.1 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>

### 8.264.2 Field Documentation

- 8.264.2.1 **BYTE** `IndFieldsList::indicationFields`[256]
- 8.264.2.2 **BYTE** `IndFieldsList::indicationFieldsLen`

## 8.265 infoInterFreq Struct Reference

### Data Fields

- [WORD](#) earfcn
- [BYTE](#) threshXLow
- [BYTE](#) threshXHigh
- [BYTE](#) cell\_resel\_priority
- [BYTE](#) cells\_len
- [cellParams](#) cellInterFreqParams [255]

### 8.265.1 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>cellInterFreq- Params[MAX_D- ESCRIPTION_L- ENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cellParams</a> for more information.</li> </ul>

### 8.265.2 Field Documentation

8.265.2.1 **BYTE** infoInterFreq::cell\_resel\_priority

8.265.2.2 **cellParams** infoInterFreq::cellInterFreqParams[255]

8.265.2.3 **BYTE** infoInterFreq::cells\_len

8.265.2.4 WORD infoInterFreq::earfcn

8.265.2.5 BYTE infoInterFreq::threshXHigh

8.265.2.6 BYTE infoInterFreq::threshXLow

## 8.266 IOThresh Struct Reference

### Data Fields

- [BYTE IOThresListLen](#)
- [INT32 \\* pIOThresList](#)

### 8.266.1 Detailed Description

This structure contains IO threshold related parameters.

#### Parameters

<i>IOThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
<i>pIOThresList</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>

### 8.266.2 Field Documentation

8.266.2.1 BYTE IOThresh::IOThresListLen

8.266.2.2 INT32\* IOThresh::pIOThresList

## 8.267 IPv4Addr Struct Reference

### Data Fields

- [ULONG addr](#)
- [ULONG subnetMask](#)

### 8.267.1 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>

## 8.267.2 Field Documentation

## 8.267.2.1 ULONG IPv4Addr::addr

## 8.267.2.2 ULONG IPv4Addr::subnetMask

## 8.268 IPv6Addr Struct Reference

## Data Fields

- [BYTE addr](#) [16]
- [BYTE prefixLen](#)

## 8.268.1 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.

## 8.268.2 Field Documentation

## 8.268.2.1 BYTE IPv6Addr::addr[16]

## 8.268.2.2 BYTE IPv6Addr::prefixLen

## 8.269 IPV6AddressInfo Struct Reference

## Data Fields

- [BYTE IPV6PrefixLen](#)
- [USHORT IPAddressV6](#) [8]

## 8.269.1 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>

## 8.269.2 Field Documentation

8.269.2.1 USHORT IPV6AddressInfo::IPAddressV6[8]

8.269.2.2 BYTE IPV6AddressInfo::IPV6PrefixLen

## 8.270 ipv6AddressInfo Struct Reference

## Data Fields

- uint8\_t [IPV6PrefixLen](#)
- uint16\_t [IPAddressV6](#) [8]

## 8.270.1 Detailed Description

## Parameters

<i>IPV6PrefixLen</i>	Length of the received IPv6 address
<i>IPAddressV6</i>	IPv6 address(in network byte order)

## 8.270.2 Field Documentation

8.270.2.1 uint16\_t ipv6AddressInfo::IPAddressV6[8]

8.270.2.2 uint8\_t ipv6AddressInfo::IPV6PrefixLen

## 8.271 IPV6GWAddressInfo Struct Reference

## Data Fields

- BYTE [gwV6PrefixLen](#)
- USHORT [gwAddressV6](#) [8]

## 8.271.1 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>
--------------------	---

## 8.271.2 Field Documentation

8.271.2.1 USHORT IPv6GWAddressInfo::gwAddressV6[8]

8.271.2.2 BYTE IPv6GWAddressInfo::gwV6PrefixLen

## 8.272 IPv6TrafCls Struct Reference

### Data Fields

- [BYTE val](#)
- [BYTE mask](#)

### 8.272.1 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) 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>

## 8.272.2 Field Documentation

8.272.2.1 BYTE IPv6TrafCls::mask

8.272.2.2 BYTE IPv6TrafCls::val

## 8.273 LibPackGPRSRequestedQoS Struct Reference

### Data Fields

- uint32\_t [precedenceClass](#)
- uint32\_t [delayClass](#)
- uint32\_t [reliabilityClass](#)
- uint32\_t [peakThroughputClass](#)
- uint32\_t [meanThroughputClass](#)

### 8.273.1 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>

### 8.273.2 Field Documentation

8.273.2.1 `uint32_t LibPackGPRSRequestedQoS::delayClass`

8.273.2.2 `uint32_t LibPackGPRSRequestedQoS::meanThroughputClass`

8.273.2.3 `uint32_t LibPackGPRSRequestedQoS::peakThroughputClass`

8.273.2.4 `uint32_t LibPackGPRSRequestedQoS::precedenceClass`

8.273.2.5 `uint32_t LibPackGPRSRequestedQoS::reliabilityClass`

## 8.274 LibpackProfile3GPP Struct Reference

### Data Fields

- `uint8_t * pProfilename`
- `uint16_t * pProfilenameSize`
- `uint8_t * pPDPTtype`
- `uint8_t * pPdpHdrCompType`
- `uint8_t * pPdpDataCompType`
- `uint8_t * pAPNName`
- `uint16_t * pAPNnameSize`
- `uint32_t * pPriDNSIPv4AddPref`
- `uint32_t * pSecDNSIPv4AddPref`
- `LibPackUMTSQoS * pUMTSReqQoS`
- `LibPackUMTSQoS * pUMTSMinQoS`
- `LibPackGPRSRequestedQoS * pGPRSRequestedQoS`
- `LibPackGPRSRequestedQoS * pGPRSMinimumQoS`
- `uint8_t * pUsername`
- `uint16_t * pUsernameSize`
- `uint8_t * pPassword`

- uint16\_t \* pPasswordSize
- uint8\_t \* pAuthenticationPref
- uint32\_t \* pIPv4AddrPref
- uint8\_t \* pPcscfAddrUsingPCO
- uint8\_t \* pPdpAccessConFlag
- uint8\_t \* pPcscfAddrUsingDhcp
- uint8\_t \* pImCnFlag
- LibPackTFTIDParams \* pTFTID1Params
- LibPackTFTIDParams \* pTFTID2Params
- uint8\_t \* pPdpContext
- uint8\_t \* pSecondaryFlag
- uint8\_t \* pPrimaryID
- uint16\_t \* pIPv6AddPref
- LibPackUMTSReqQoSsigInd \* pUMTSReqQoSsigInd
- LibPackUMTSReqQoSsigInd \* pUMTSMInQoSsigInd
- uint16\_t \* pPriDNSIPv6addpref
- uint16\_t \* pSecDNSIPv6addpref
- uint8\_t \* pAddrAllocPref
- LibPackQoSClassID \* pQoSClassID
- uint8\_t \* pAPNDisabledFlag
- uint32\_t \* pPDNInactivTimeout
- uint8\_t \* pAPNClass

### 8.274.1 Detailed Description

#### Parameters

<i>extended</i>	error
<i>profile</i>	<p>info This structure contains Input parameters of SLQSCreateProfile and SLQSModifyProfile and output parameters of SLQSGetProfileSettings</p> <ul style="list-style-type: none"> <li>• Parameter values default to their data type's maximum unsigned value unless explicitly stated otherwise.</li> </ul>
<i>pProfileName</i>	<ul style="list-style-type: none"> <li>• One or more uint8_ts 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 pProfileName field. Size of this parameter is 2 uint8_ts.</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>pPdpDataCompType</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 pAPNName field. Size of this parameter is 2 uint8_ts.</li> </ul>
<i>pPriDNSIPv4AddPref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address Preference</li> </ul>
<i>pSecDNSIPv4AddPref</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>pGPRSRequestedQoS</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 uint8_ts.</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 uint8_ts.</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>plmCnFlag</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 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pPDNInactiv-Timeout</i>	<ul style="list-style-type: none"> <li>Optional 4 uint8_ts 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="#">SLQSGetProfileSettings()</a>.</li> </ul>

<i>pAPNClass</i>	<ul style="list-style-type: none"> <li>• Optional 1 uint8_t 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>
------------------	---

## 8.274.2 Field Documentation

- 8.274.2.1 uint8\_t\* LibpackProfile3GPP::pAddrAllocPref
- 8.274.2.2 uint8\_t\* LibpackProfile3GPP::pAPNClass
- 8.274.2.3 uint8\_t\* LibpackProfile3GPP::pAPNDisabledFlag
- 8.274.2.4 uint8\_t\* LibpackProfile3GPP::pAPNName
- 8.274.2.5 uint16\_t\* LibpackProfile3GPP::pAPNNameSize
- 8.274.2.6 uint8\_t\* LibpackProfile3GPP::pAuthenticationPref
- 8.274.2.7 LibPackGPRSRequestedQoS\* LibpackProfile3GPP::pGPRSMinimumQoS
- 8.274.2.8 LibPackGPRSRequestedQoS\* LibpackProfile3GPP::pGPRSRequestedQos
- 8.274.2.9 uint8\_t\* LibpackProfile3GPP::pImCnFlag
- 8.274.2.10 uint32\_t\* LibpackProfile3GPP::pIPv4AddrPref
- 8.274.2.11 uint16\_t\* LibpackProfile3GPP::pIPv6AddPref
- 8.274.2.12 uint8\_t\* LibpackProfile3GPP::pPassword
- 8.274.2.13 uint16\_t\* LibpackProfile3GPP::pPasswordSize
- 8.274.2.14 uint8\_t\* LibpackProfile3GPP::pPcsfAddrUsingDhcp
- 8.274.2.15 uint8\_t\* LibpackProfile3GPP::pPcsfAddrUsingPCO
- 8.274.2.16 uint32\_t\* LibpackProfile3GPP::pPDNInactivTimeout
- 8.274.2.17 uint8\_t\* LibpackProfile3GPP::pPdpAccessConFlag
- 8.274.2.18 uint8\_t\* LibpackProfile3GPP::pPdpContext
- 8.274.2.19 uint8\_t\* LibpackProfile3GPP::pPdpDataCompType
- 8.274.2.20 uint8\_t\* LibpackProfile3GPP::pPdpHdrCompType
- 8.274.2.21 uint8\_t\* LibpackProfile3GPP::pPDPTtype
- 8.274.2.22 uint32\_t\* LibpackProfile3GPP::pPriDNSIPv4AddPref
- 8.274.2.23 uint16\_t\* LibpackProfile3GPP::pPriDNSIPv6addpref

- 8.274.2.24 `uint8_t*` LibpackProfile3GPP::pPrimaryID
- 8.274.2.25 `uint8_t*` LibpackProfile3GPP::pProfilename
- 8.274.2.26 `uint16_t*` LibpackProfile3GPP::pProfilenameSize
- 8.274.2.27 `LibPackQosClassID*` LibpackProfile3GPP::pQosClassID
- 8.274.2.28 `uint32_t*` LibpackProfile3GPP::pSecDNSIPv4AddPref
- 8.274.2.29 `uint16_t*` LibpackProfile3GPP::pSecDNSIPv6addpref
- 8.274.2.30 `uint8_t*` LibpackProfile3GPP::pSecondaryFlag
- 8.274.2.31 `LibPackTFTIDParams*` LibpackProfile3GPP::pTFTID1Params
- 8.274.2.32 `LibPackTFTIDParams*` LibpackProfile3GPP::pTFTID2Params
- 8.274.2.33 `LibPackUMTSQoS*` LibpackProfile3GPP::pUMTSMinQoS
- 8.274.2.34 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPP::pUMTSMinQoSsigInd
- 8.274.2.35 `LibPackUMTSQoS*` LibpackProfile3GPP::pUMTSReqQoS
- 8.274.2.36 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPP::pUMTSReqQoSsigInd
- 8.274.2.37 `uint8_t*` LibpackProfile3GPP::pUsername
- 8.274.2.38 `uint16_t*` LibpackProfile3GPP::pUsernameSize

## 8.275 LibpackProfile3GPP2 Struct Reference

### Data Fields

- `uint8_t *` [pNegoDnsSrvrPref](#)
- `uint32_t *` [pPppSessCloseTimerDO](#)
- `uint32_t *` [pPppSessCloseTimer1x](#)
- `uint8_t *` [pAllowLinger](#)
- `uint16_t *` [pLcpAckTimeout](#)
- `uint16_t *` [pIpccpAckTimeout](#)
- `uint16_t *` [pAuthTimeout](#)
- `uint8_t *` [pLcpCreqRetryCount](#)
- `uint8_t *` [pIpccpCreqRetryCount](#)
- `uint8_t *` [pAuthRetryCount](#)
- `uint8_t *` [pAuthProtocol](#)
- `uint8_t *` [pUserId](#)
- `uint16_t *` [pUserIdSize](#)
- `uint8_t *` [pAuthPassword](#)
- `uint16_t *` [pAuthPasswordSize](#)
- `uint8_t *` [pDataRate](#)
- `uint32_t *` [pAppType](#)
- `uint8_t *` [pDataMode](#)
- `uint8_t *` [pAppPriority](#)
- `uint8_t *` [pApnString](#)
- `uint16_t *` [pApnStringSize](#)

- uint8\_t \* [pPdnType](#)
- uint8\_t \* [plsPcscfAddressNedded](#)
- uint32\_t \* [pPrimaryV4DnsAddress](#)
- uint32\_t \* [pSecondaryV4DnsAddress](#)
- uint16\_t \* [pPriV6DnsAddress](#)
- uint16\_t \* [pSecV6DnsAddress](#)
- uint8\_t \* [pRATType](#)
- uint8\_t \* [pAPNEnabled3GPP2](#)
- uint32\_t \* [pPDNInactivTimeout3GPP2](#)
- uint8\_t \* [pAPNClass3GPP2](#)

### 8.275.1 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 uint8_ts;</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 uint8_ts.</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 uint8_ts</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 uint8_ts.</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 uint8_t 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 uint8_ts</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 uint8_ts.</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 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pAPNEnabled3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pPDNInactivityTimeout3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 4 uint8_ts 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pAPNClass3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t 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>

## 8.275.2 Field Documentation

8.275.2.1 uint8\_t\* LibpackProfile3GPP2::pAllowLinger

8.275.2.2 uint8\_t\* LibpackProfile3GPP2::pAPNClass3GPP2

8.275.2.3 uint8\_t\* LibpackProfile3GPP2::pAPNEnabled3GPP2

8.275.2.4 uint8\_t\* LibpackProfile3GPP2::pApnString

8.275.2.5 uint16\_t\* LibpackProfile3GPP2::pApnStringSize

8.275.2.6 uint8\_t\* LibpackProfile3GPP2::pAppPriority

8.275.2.7 uint32\_t\* LibpackProfile3GPP2::pAppType

8.275.2.8 uint8\_t\* LibpackProfile3GPP2::pAuthPassword

8.275.2.9 uint16\_t\* LibpackProfile3GPP2::pAuthPasswordSize

8.275.2.10 uint8\_t\* LibpackProfile3GPP2::pAuthProtocol

8.275.2.11 uint8\_t\* LibpackProfile3GPP2::pAuthRetryCount

8.275.2.12 uint16\_t\* LibpackProfile3GPP2::pAuthTimeout

8.275.2.13 uint8\_t\* LibpackProfile3GPP2::pDataMode

8.275.2.14 uint8\_t\* LibpackProfile3GPP2::pDataRate

8.275.2.15 uint16\_t\* LibpackProfile3GPP2::plpcpAckTimeout

8.275.2.16 uint8\_t\* LibpackProfile3GPP2::plpcpCreqRetryCount

8.275.2.17 uint8\_t\* LibpackProfile3GPP2::plsPcscfAddressNedded

8.275.2.18 uint16\_t\* LibpackProfile3GPP2::pLcpAckTimeout

8.275.2.19 uint8\_t\* LibpackProfile3GPP2::pLcpCreqRetryCount

8.275.2.20 uint8\_t\* LibpackProfile3GPP2::pNegoDnsSrvrPref

8.275.2.21 uint32\_t\* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2

8.275.2.22 uint8\_t\* LibpackProfile3GPP2::pPdnType

8.275.2.23 uint32\_t\* LibpackProfile3GPP2::pPppSessCloseTimer1x

8.275.2.24 uint32\_t\* LibpackProfile3GPP2::pPppSessCloseTimerDO

8.275.2.25 uint32\_t\* LibpackProfile3GPP2::pPrimaryV4DnsAddress

8.275.2.26 uint16\_t\* LibpackProfile3GPP2::pPriV6DnsAddress

8.275.2.27 uint8\_t\* LibpackProfile3GPP2::pRATType

8.275.2.28 uint32\_t\* LibpackProfile3GPP2::pSecondaryV4DnsAddress

8.275.2.29 uint16\_t\* LibpackProfile3GPP2::pSecV6DnsAddress

8.275.2.30 uint8\_t\* LibpackProfile3GPP2::pUserId

8.275.2.31 uint16\_t\* LibpackProfile3GPP2::pUserIdSize

## 8.276 LibPackprofile\_3GPP Struct Reference

### Data Fields

- uint8\_t \* [pProfilename](#)
- uint16\_t \* [pProfilenameSize](#)
- uint8\_t \* [pPDPTtype](#)
- uint8\_t \* [pPdpHdrCompType](#)
- uint8\_t \* [pPdpDataCompType](#)
- uint8\_t \* [pAPNName](#)
- uint16\_t \* [pAPNnameSize](#)
- uint32\_t \* [pPriDNSIPv4AddPref](#)

- uint32\_t \* pSecDNSIPv4AddPref
- LibPackUMTSQoS \* pUMTSReqQoS
- LibPackUMTSQoS \* pUMTSMinQoS
- LibPackGPRSRequestedQoS \* pGPRSRequestedQoS
- LibPackGPRSRequestedQoS \* pGPRSMinimumQoS
- uint8\_t \* pUsername
- uint16\_t \* pUsernameSize
- uint8\_t \* pPassword
- uint16\_t \* pPasswordSize
- uint8\_t \* pAuthenticationPref
- uint32\_t \* pIPv4AddrPref
- uint8\_t \* pPcscfAddrUsingPCO
- uint8\_t \* pPdpAccessConFlag
- uint8\_t \* pPcscfAddrUsingDhcp
- uint8\_t \* pImCnFlag
- LibPackTFTIDParams \* pTFTID1Params
- LibPackTFTIDParams \* pTFTID2Params
- uint8\_t \* pPdpContext
- uint8\_t \* pSecondaryFlag
- uint8\_t \* pPrimaryID
- uint16\_t \* pIPv6AddPref
- LibPackUMTSReqQoSSigInd \* pUMTSReqQoSSigInd
- LibPackUMTSReqQoSSigInd \* pUMTSMinQoSsigInd
- uint16\_t \* pPriDNSIPv6addpref
- uint16\_t \* pSecDNSIPv6addpref
- uint8\_t \* pAddrAllocPref
- LibPackQoSClassID \* pQoSClassID
- uint8\_t \* pAPNDisabledFlag
- uint32\_t \* pPDNInactivTimeout
- uint8\_t \* pAPNClass

### 8.276.1 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 pProfileName field. Size of this parameter is 2 bytes.</li> </ul>
<i>pPDPTYPE</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 pAPNName 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>plmCnFlag</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 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pPDNInactivity-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="#">SLQSGetProfileSettings()</a>.</li> </ul>

<i>pAPNClass</i>	<ul style="list-style-type: none"> <li>• Optional 1 uint8_t 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>
------------------	---

## 8.276.2 Field Documentation

- 8.276.2.1 uint8\_t\* LibPackprofile\_3GPP::pAddrAllocPref
- 8.276.2.2 uint8\_t\* LibPackprofile\_3GPP::pAPNClass
- 8.276.2.3 uint8\_t\* LibPackprofile\_3GPP::pAPNDisabledFlag
- 8.276.2.4 uint8\_t\* LibPackprofile\_3GPP::pAPNName
- 8.276.2.5 uint16\_t\* LibPackprofile\_3GPP::pAPNnameSize
- 8.276.2.6 uint8\_t\* LibPackprofile\_3GPP::pAuthenticationPref
- 8.276.2.7 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSMinimumQoS
- 8.276.2.8 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSRequestedQoS
- 8.276.2.9 uint8\_t\* LibPackprofile\_3GPP::plmCnFlag
- 8.276.2.10 uint32\_t\* LibPackprofile\_3GPP::pIPv4AddrPref
- 8.276.2.11 uint16\_t\* LibPackprofile\_3GPP::pIPv6AddPref
- 8.276.2.12 uint8\_t\* LibPackprofile\_3GPP::pPassword
- 8.276.2.13 uint16\_t\* LibPackprofile\_3GPP::pPasswordSize
- 8.276.2.14 uint8\_t\* LibPackprofile\_3GPP::pPcscfAddrUsingDhcp
- 8.276.2.15 uint8\_t\* LibPackprofile\_3GPP::pPcscfAddrUsingPCO
- 8.276.2.16 uint32\_t\* LibPackprofile\_3GPP::pPDNInactivTimeout
- 8.276.2.17 uint8\_t\* LibPackprofile\_3GPP::pPdpAccessConFlag
- 8.276.2.18 uint8\_t\* LibPackprofile\_3GPP::pPdpContext
- 8.276.2.19 uint8\_t\* LibPackprofile\_3GPP::pPdpDataCompType
- 8.276.2.20 uint8\_t\* LibPackprofile\_3GPP::pPdpHdrCompType
- 8.276.2.21 uint8\_t\* LibPackprofile\_3GPP::pPDPTtype
- 8.276.2.22 uint32\_t\* LibPackprofile\_3GPP::pPriDNSIPv4AddPref
- 8.276.2.23 uint16\_t\* LibPackprofile\_3GPP::pPriDNSIPv6addpref

- 8.276.2.24 `uint8_t*` `LibPackprofile_3GPP::pPrimaryID`
- 8.276.2.25 `uint8_t*` `LibPackprofile_3GPP::pProfileName`
- 8.276.2.26 `uint16_t*` `LibPackprofile_3GPP::pProfileNameSize`
- 8.276.2.27 `LibPackQosClassID*` `LibPackprofile_3GPP::pQosClassID`
- 8.276.2.28 `uint32_t*` `LibPackprofile_3GPP::pSecDNSIPv4AddPref`
- 8.276.2.29 `uint16_t*` `LibPackprofile_3GPP::pSecDNSIPv6addpref`
- 8.276.2.30 `uint8_t*` `LibPackprofile_3GPP::pSecondaryFlag`
- 8.276.2.31 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID1Params`
- 8.276.2.32 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID2Params`
- 8.276.2.33 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSMinQoS`
- 8.276.2.34 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSMinQoSsigInd`
- 8.276.2.35 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSReqQoS`
- 8.276.2.36 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSReqQoSSigInd`
- 8.276.2.37 `uint8_t*` `LibPackprofile_3GPP::pUsername`
- 8.276.2.38 `uint16_t*` `LibPackprofile_3GPP::pUsernameSize`

## 8.277 LibPackprofile\_3GPP2 Struct Reference

### Data Fields

- `uint8_t *` [pNegoDnsSrvrPref](#)
- `uint32_t *` [pPppSessCloseTimerDO](#)
- `uint32_t *` [pPppSessCloseTimer1x](#)
- `uint8_t *` [pAllowLinger](#)
- `uint16_t *` [pLcpAckTimeout](#)
- `uint16_t *` [pIpccpAckTimeout](#)
- `uint16_t *` [pAuthTimeout](#)
- `uint8_t *` [pLcpCreqRetryCount](#)
- `uint8_t *` [pIpccpCreqRetryCount](#)
- `uint8_t *` [pAuthRetryCount](#)
- `uint8_t *` [pAuthProtocol](#)
- `uint8_t *` [pUserId](#)
- `uint16_t *` [pUserIdSize](#)
- `uint8_t *` [pAuthPassword](#)
- `uint16_t *` [pAuthPassword\\_tSize](#)
- `uint8_t *` [pDataRate](#)
- `uint32_t *` [pAppType](#)
- `uint8_t *` [pDataMode](#)
- `uint8_t *` [pAppPriority](#)
- `uint8_t *` [pApnString](#)
- `uint16_t *` [pApnStringSize](#)

- uint8\_t \* [pPdnType](#)
- uint8\_t \* [plsPcscfAddressNedded](#)
- uint32\_t \* [pPrimaryV4DnsAddress](#)
- uint32\_t \* [pSecondaryV4DnsAddress](#)
- uint16\_t \* [pPriV6DnsAddress](#)
- uint16\_t \* [pSecV6DnsAddress](#)
- uint8\_t \* [pRATType](#)
- uint8\_t \* [pAPNEnabled3GPP2](#)
- uint32\_t \* [pPDNInactivTimeout3GPP2](#)
- uint8\_t \* [pAPNClass3GPP2](#)

### 8.277.1 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 uint8_t 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 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pAPNEnabled3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t 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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pPDNInactivityTimeout3GPP2</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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pAPNClass3GPP2</i>	<ul style="list-style-type: none"> <li>Optional 1 uint8_t 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>

## 8.277.2 Field Documentation

8.277.2.1 uint8\_t\* LibPackprofile\_3GPP2::pAllowLinger

8.277.2.2 uint8\_t\* LibPackprofile\_3GPP2::pAPNClass3GPP2

8.277.2.3 uint8\_t\* LibPackprofile\_3GPP2::pAPNEnabled3GPP2

8.277.2.4 uint8\_t\* LibPackprofile\_3GPP2::pApnString

8.277.2.5 uint16\_t\* LibPackprofile\_3GPP2::pApnStringSize

8.277.2.6 uint8\_t\* LibPackprofile\_3GPP2::pAppPriority

8.277.2.7 uint32\_t\* LibPackprofile\_3GPP2::pAppType

8.277.2.8 uint8\_t\* LibPackprofile\_3GPP2::pAuthPassword

8.277.2.9 uint16\_t\* LibPackprofile\_3GPP2::pAuthPassword\_tSize

- 8.277.2.10 uint8\_t\* LibPackprofile\_3GPP2::pAuthProtocol
- 8.277.2.11 uint8\_t\* LibPackprofile\_3GPP2::pAuthRetryCount
- 8.277.2.12 uint16\_t\* LibPackprofile\_3GPP2::pAuthTimeout
- 8.277.2.13 uint8\_t\* LibPackprofile\_3GPP2::pDataMode
- 8.277.2.14 uint8\_t\* LibPackprofile\_3GPP2::pDataRate
- 8.277.2.15 uint16\_t\* LibPackprofile\_3GPP2::plpcpAckTimeout
- 8.277.2.16 uint8\_t\* LibPackprofile\_3GPP2::plpcpCreqRetryCount
- 8.277.2.17 uint8\_t\* LibPackprofile\_3GPP2::plsPcscfAddressNedded
- 8.277.2.18 uint16\_t\* LibPackprofile\_3GPP2::pLcpAckTimeout
- 8.277.2.19 uint8\_t\* LibPackprofile\_3GPP2::pLcpCreqRetryCount
- 8.277.2.20 uint8\_t\* LibPackprofile\_3GPP2::pNegoDnsSrvrPref
- 8.277.2.21 uint32\_t\* LibPackprofile\_3GPP2::pPDNInactivTimeout3GPP2
- 8.277.2.22 uint8\_t\* LibPackprofile\_3GPP2::pPdnType
- 8.277.2.23 uint32\_t\* LibPackprofile\_3GPP2::pPppSessCloseTimer1x
- 8.277.2.24 uint32\_t\* LibPackprofile\_3GPP2::pPppSessCloseTimerDO
- 8.277.2.25 uint32\_t\* LibPackprofile\_3GPP2::pPrimaryV4DnsAddress
- 8.277.2.26 uint16\_t\* LibPackprofile\_3GPP2::pPriV6DnsAddress
- 8.277.2.27 uint8\_t\* LibPackprofile\_3GPP2::pRATType
- 8.277.2.28 uint32\_t\* LibPackprofile\_3GPP2::pSecondaryV4DnsAddress
- 8.277.2.29 uint16\_t\* LibPackprofile\_3GPP2::pSecV6DnsAddress
- 8.277.2.30 uint8\_t\* LibPackprofile\_3GPP2::pUserId
- 8.277.2.31 uint16\_t\* LibPackprofile\_3GPP2::pUserIdSize

## 8.278 LibPackQosClassID Struct Reference

### Data Fields

- uint8\_t [QCI](#)
- uint8\_t [gDIBitRate](#)
- uint32\_t [maxDIBitRate](#)
- uint32\_t [gUIBitRate](#)
- uint32\_t [maxUIBitRate](#)

### 8.278.1 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>

### 8.278.2 Field Documentation

8.278.2.1 `uint8_t LibPackQosClassID::gDlBitRate`

8.278.2.2 `uint32_t LibPackQosClassID::gUlBitRate`

8.278.2.3 `uint32_t LibPackQosClassID::maxDlBitRate`

8.278.2.4 `uint32_t LibPackQosClassID::maxUlBitRate`

8.278.2.5 `uint8_t LibPackQosClassID::QCI`

## 8.279 LibPackTFTIDParams Struct Reference

### Data Fields

- `uint8_t filterId`
- `uint8_t eValid`
- `uint8_t ipVersion`
- `uint16_t * pSourceIP`
- `uint8_t sourceIPMask`
- `uint8_t nextHeader`
- `uint32_t destPortRangeStart`
- `uint16_t destPortRangeEnd`
- `uint16_t srcPortRangeStart`
- `uint16_t srcPortRangeEnd`
- `uint32_t IPSECSPi`
- `uint16_t tosMask`
- `uint32_t flowLabel`

### 8.279.1 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 uint8_ts</li> <li>– IPv6 - Fill all the 16 uint8_ts</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>

### 8.279.2 Field Documentation

- 8.279.2.1    `uint16_t LibPackTFTIDParams::destPortRangeEnd`
- 8.279.2.2    `uint32_t LibPackTFTIDParams::destPortRangeStart`
- 8.279.2.3    `uint8_t LibPackTFTIDParams::eValid`
- 8.279.2.4    `uint8_t LibPackTFTIDParams::filterId`
- 8.279.2.5    `uint32_t LibPackTFTIDParams::flowLabel`
- 8.279.2.6    `uint32_t LibPackTFTIDParams::IPSECSPi`
- 8.279.2.7    `uint8_t LibPackTFTIDParams::ipVersion`
- 8.279.2.8    `uint8_t LibPackTFTIDParams::nextHeader`
- 8.279.2.9    `uint16_t* LibPackTFTIDParams::pSourceIP`
- 8.279.2.10   `uint8_t LibPackTFTIDParams::sourceIPMask`
- 8.279.2.11   `uint16_t LibPackTFTIDParams::srcPortRangeEnd`
- 8.279.2.12   `uint16_t LibPackTFTIDParams::srcPortRangeStart`
- 8.279.2.13   `uint16_t LibPackTFTIDParams::tosMask`

## 8.280    LibPackUMTSQoS Struct Reference

### Data Fields

- `uint8_t trafficClass`
- `uint32_t maxUplinkBitrate`
- `uint32_t maxDownlinkBitrate`
- `uint32_t grntUplinkBitrate`
- `uint32_t grntDownlinkBitrate`
- `uint8_t qosDeliveryOrder`
- `uint32_t maxSDUSize`
- `uint8_t sduErrorRatio`
- `uint8_t resBerRatio`
- `uint8_t deliveryErrSDU`
- `uint32_t transferDelay`
- `uint32_t trafficPriority`

### 8.280.1   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>maxDownlinkBitrate</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>grntDownlinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDeliveryOrder</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>	<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 \times 10^{-2}</math></li> <li>• 0x02 - <math>1 \times 10^{-2}</math></li> <li>• 0x03 - <math>5 \times 10^{-3}</math></li> <li>• 0x04 - <math>4 \times 10^{-3}</math></li> <li>• 0x05 - <math>1 \times 10^{-3}</math></li> <li>• 0x06 - <math>1 \times 10^{-4}</math></li> <li>• 0x07 - <math>1 \times 10^{-5}</math></li> <li>• 0x08 - <math>1 \times 10^{-6}</math></li> <li>• 0x09 - <math>1 \times 10^{-8}</math></li> </ul>

<i>deliveryErrSDU</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>transferDelay</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>trafficPriority</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>

## 8.280.2 Field Documentation

8.280.2.1 `uint8_t LibPackUMTSQoS::deliveryErrSDU`

8.280.2.2 `uint32_t LibPackUMTSQoS::grntDownlinkBitrate`

8.280.2.3 `uint32_t LibPackUMTSQoS::grntUplinkBitrate`

8.280.2.4 `uint32_t LibPackUMTSQoS::maxDownlinkBitrate`

8.280.2.5 `uint32_t LibPackUMTSQoS::maxSDUSize`

8.280.2.6 `uint32_t LibPackUMTSQoS::maxUplinkBitrate`

8.280.2.7 `uint8_t LibPackUMTSQoS::qosDeliveryOrder`

8.280.2.8 `uint8_t LibPackUMTSQoS::resBerRatio`

8.280.2.9 `uint8_t LibPackUMTSQoS::sduErrorRatio`

8.280.2.10 `uint8_t LibPackUMTSQoS::trafficClass`

8.280.2.11 `uint32_t LibPackUMTSQoS::trafficPriority`

8.280.2.12 `uint32_t LibPackUMTSQoS::transferDelay`

## 8.281 LibPackUMTSReqQoSsigInd Struct Reference

## Data Fields

- [LibPackUMTSQoS UMTSReqQoS](#)
- [uint8\\_t SigInd](#)

### 8.281.1 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>

### 8.281.2 Field Documentation

8.281.2.1 [uint8\\_t LibPackUMTSReqQoSSigInd::SigInd](#)

8.281.2.2 [LibPackUMTSQoS LibPackUMTSReqQoSSigInd::UMTSReqQoS](#)

## 8.282 lineCtrlInfo Struct Reference

## Data Fields

- [BYTE polarityIncluded](#)
- [BYTE toggleMode](#)
- [BYTE revPolarity](#)
- [BYTE pwrDenialTime](#)

### 8.282.1 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>

### 8.282.2 Field Documentation

8.282.2.1 BYTE lineCtrlInfo::polarityIncluded

8.282.2.2 BYTE lineCtrlInfo::pwrDenialTime

8.282.2.3 BYTE lineCtrlInfo::revPolarity

8.282.2.4 BYTE lineCtrlInfo::toggleMode

## 8.283 loc\_BdsSV Struct Reference

### Data Fields

- uint16\_t [id](#)
- uint8\_t [mask](#)

### 8.283.1 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>

### 8.283.2 Field Documentation

8.283.2.1 uint16\_t loc\_BdsSV::id

8.283.2.2 uint8\_t loc\_BdsSV::mask

## 8.284 loc\_BdsSVInfo Struct Reference

### Data Fields

- uint8\_t [len](#)
- [loc\\_BdsSV](#) \* [pSV](#)

### 8.284.1 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="#">loc_BdsSV</a>. See <a href="#">loc_BdsSV</a> for more information</li> </ul>

## 8.284.2 Field Documentation

8.284.2.1 uint8\_t loc\_BdsSVInfo::len

8.284.2.2 loc\_BdsSV\* loc\_BdsSVInfo::pSV

## 8.285 loc\_CellDb Struct Reference

## Data Fields

- uint32\_t [mask](#)

## 8.285.1 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>
-------------	--

## 8.285.2 Field Documentation

8.285.2.1 uint32\_t loc\_CellDb::mask

## 8.286 loc\_ClkInfo Struct Reference

## Data Fields

- uint32\_t [mask](#)

### 8.286.1 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>
-------------	--

### 8.286.2 Field Documentation

8.286.2.1    `uint32_t loc_ClkInfo::mask`

## 8.287    loc\_GnssData Struct Reference

### Data Fields

- `uint64_t` [mask](#)

### 8.287.1    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 GLONASS SVDIR</li> <li>– QMI_LOC_MASK_DELETE_GLO_SVSTEER (0x00000020) - Mask to delete GLONASS 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>
-------------	---

## 8.287.2 Field Documentation

### 8.287.2.1 uint64\_t loc\_GnssData::mask

## 8.288 loc\_gpsTime Struct Reference

## Data Fields

- uint16\_t [gpsWeek](#)
- uint32\_t [gpsTimeOfWeekMs](#)

## 8.288.1 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>

## 8.288.2 Field Documentation

8.288.2.1 uint32\_t loc\_gpsTime::gpsTimeOfWeekMs

8.288.2.2 uint16\_t loc\_gpsTime::gpsWeek

## 8.289 loc\_LocApplicationInfo Struct Reference

## Data Fields

- uint8\_t [appProviderLength](#)
- uint8\_t \* [pAppProvider](#)
- uint8\_t [appNameLength](#)
- uint8\_t \* [pAppName](#)
- uint8\_t [appVersionValid](#)
- uint8\_t [appVersionLength](#)
- uint8\_t \* [pAppVersion](#)

## 8.289.1 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>appVersion- Length</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>

## 8.289.2 Field Documentation

8.289.2.1 `uint8_t loc_LocApplicationInfo::appNameLength`

8.289.2.2 `uint8_t loc_LocApplicationInfo::appProviderLength`

8.289.2.3 `uint8_t loc_LocApplicationInfo::appVersionLength`

8.289.2.4 `uint8_t loc_LocApplicationInfo::appVersionValid`

8.289.2.5 `uint8_t* loc_LocApplicationInfo::pAppName`

8.289.2.6 `uint8_t* loc_LocApplicationInfo::pAppProvider`

8.289.2.7 `uint8_t* loc_LocApplicationInfo::pAppVersion`

## 8.290 loc\_precisionDilution Struct Reference

### Data Fields

- `uint32_t` [PDOP](#)
- `uint32_t` [HDOP](#)
- `uint32_t` [VDOP](#)

### 8.290.1 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>

## 8.290.2 Field Documentation

8.290.2.1 uint32\_t loc\_precisionDilution::HDOP

8.290.2.2 uint32\_t loc\_precisionDilution::PDOP

8.290.2.3 uint32\_t loc\_precisionDilution::VDOP

## 8.291 loc\_sensorDataUsage Struct Reference

### Data Fields

- uint32\_t [usageMask](#)
- uint32\_t [aidingIndicatorMask](#)

### 8.291.1 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>	
----------------------------	--

- Value
  - Specifies which results were aided by sensors.

- Value
  - 0x00000001 - AIDED\_HEADING
  - 0x00000002 - AIDED\_SPEED
  - 0x00000004 - AIDED\_POSITION
  - 0x00000008 - AIDED\_VELOCITY

## 8.291.2 Field Documentation

8.291.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.291.2.2 `uint32_t loc_sensorDataUsage::usageMask`

## 8.292 loc\_SV Struct Reference

### Data Fields

- `uint16_t id`
- `uint32_t system`
- `uint8_t mask`

### 8.292.1 Detailed Description

This structure contains the Delete LOC SV Info

#### Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>• LOC SV 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 loc_SV belongs</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– <code>eQMI_LOC_SV_SYSTEM_GPS</code> (1) - GPS satellite</li> <li>– <code>eQMI_LOC_SV_SYSTEM_GALILEO</code> (2) - GALILEO satellite</li> <li>– <code>eQMI_LOC_SV_SYSTEM_SBAS</code> (3) - SBAS satellite</li> <li>– <code>eQMI_LOC_SV_SYSTEM_COMPASS</code> (4) - COMPASS satellite</li> <li>– <code>eQMI_LOC_SV_SYSTEM_GLONASS</code> (5) - GLONASS satellite</li> <li>– <code>eQMI_LOC_SV_SYSTEM_BDS</code> (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>– <code>0x01</code> - <code>DELETE_EPHEMERIS</code></li> <li>– <code>0x02</code> - <code>DELETE_ALMANAC</code></li> </ul> </li> </ul>

### 8.292.2 Field Documentation

8.292.2.1 `uint16_t loc_SV::id`

8.292.2.2 `uint8_t loc_SV::mask`

8.292.2.3 `uint32_t loc_SV::system`

## 8.293 loc\_SVInfo Struct Reference

## Data Fields

- uint8\_t [len](#)
- [loc\\_SV](#) \* pSV

## 8.293.1 Detailed Description

This structure contains the elements of Delete LOC [SV](#) Info

## Parameters

<i>len</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements in struct <a href="#">loc_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="#">loc_SV</a>. See <a href="#">loc_SV</a> for more information</li> </ul>

## 8.293.2 Field Documentation

8.293.2.1 uint8\_t loc\_SVInfo::len

8.293.2.2 [loc\\_SV](#)\* loc\_SVInfo::pSV

## 8.294 loc\_svUsedforFix Struct Reference

## Data Fields

- uint8\_t [gnssSvUsedList\\_len](#)
- uint16\_t [gnssSvUsedList](#) [255]

## 8.294.1 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>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>

## 8.294.2 Field Documentation

8.294.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`

8.294.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`

## 8.295 LocApplicationInfo Struct Reference

### Data Fields

- [BYTE appProviderLength](#)
- [CHAR \\* pAppProvider](#)
- [BYTE appNameLength](#)
- [CHAR \\* pAppName](#)
- [BYTE appVersionValid](#)
- [CHAR appVersionLength](#)
- [CHAR \\* pAppVersion](#)

### 8.295.1 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>

### 8.295.2 Field Documentation

- 8.295.2.1 **BYTE** LocApplicationInfo::appNameLength
- 8.295.2.2 **BYTE** LocApplicationInfo::appProviderLength
- 8.295.2.3 **CHAR** LocApplicationInfo::appVersionLength
- 8.295.2.4 **BYTE** LocApplicationInfo::appVersionValid
- 8.295.2.5 **CHAR\*** LocApplicationInfo::pAppName
- 8.295.2.6 **CHAR\*** LocApplicationInfo::pAppProvider
- 8.295.2.7 **CHAR\*** LocApplicationInfo::pAppVersion

## 8.296 LocDelAssDataReq Struct Reference

### Data Fields

- [SVInfo](#) \* [pSVInfo](#)
- [GnssData](#) \* [pGnssData](#)
- [CellDb](#) \* [pCellDb](#)
- [ClkInfo](#) \* [pClkInfo](#)
- [BdsSVInfo](#) \* [pBdsSVInfo](#)

### 8.296.1 Detailed Description

This structure contains LOC delete assist data request

#### Parameters

<i>pSVInfo</i> [IN]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">SVInfo</a>. See <a href="#">SVInfo</a> for more information</li> </ul>
<i>pGnssData</i> [IN]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">GnssData</a>. See <a href="#">GnssData</a> for more information</li> </ul>
<i>pCellDb</i> [IN]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">CellDb</a>. See <a href="#">CellDb</a> for more information</li> </ul>
<i>pClkInfo</i> [IN]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">ClkInfo</a>. See <a href="#">ClkInfo</a> for more information</li> </ul>
<i>pBdsSVInfo</i> [IN]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">BdsSVInfo</a>. See <a href="#">BdsSVInfo</a> for more information</li> </ul>

### 8.296.2 Field Documentation

8.296.2.1 **BdsSVInfo\*** LocDelAssDataReq::pBdsSVInfo

8.296.2.2 **CellDb\*** LocDelAssDataReq::pCellDb

8.296.2.3 **ClkInfo\*** LocDelAssDataReq::pClkInfo

8.296.2.4 **GnssData\*** LocDelAssDataReq::pGnssData

8.296.2.5 **SVInfo\*** LocDelAssDataReq::pSVInfo

## 8.297 LOCEventRegisterReqResp Struct Reference

### Data Fields

- [ULONGLONG eventRegister](#)

### 8.297.1 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. These reports are sent at a 1 Hz rate.</li> <li>– 0x00000004 - to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.</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 engine state report event indications.</li> <li>– 0x00000100 - 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 (accelerometer, gyroscope, etc.).</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. Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG.</li> </ul> </li> </ul>
----------------------	--

**Note**

Multiple events can be registered by OR the individual masks and sending them in this TLV. All unused bits in this mask must be set to 0.

**8.297.2 Field Documentation****8.297.2.1 ULONGLONG LOCEventRegisterReqResp::eventRegister****8.298 LOCExtPowerStateReqResp Struct Reference****Data Fields**

- [ULONG extPowerState](#)

**8.298.1 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>
-----------------------------------	---

**8.298.2 Field Documentation****8.298.2.1 ULONG LOCExtPowerStateReqResp::extPowerState****8.299 LocInjectPositionReq Struct Reference****Data Fields**

- [ULONGLONG \\* pLatitude](#)
- [ULONGLONG \\* pLongitude](#)
- [ULONG \\* pHorUncCircular](#)
- [BYTE \\* pHorConfidence](#)
- [ULONG \\* pHorReliability](#)
- [ULONG \\* pAltitudeWrtEllipsoid](#)
- [ULONG \\* pAltitudeWrtMeanSeaLevel](#)
- [ULONG \\* pVertUnc](#)
- [BYTE \\* pVertConfidence](#)
- [ULONG \\* pVertReliability](#)
- [altitudeSrcInfo \\* pAltitudeSrcInfo](#)
- [ULONGLONG \\* pTimestampUtc](#)
- [ULONG \\* pTimestampAge](#)
- [ULONG \\* pPositionSrc](#)
- [ULONG \\* pRawHorUncCircular](#)
- [BYTE \\* pRawHorConfidence](#)

### 8.299.1 Detailed Description

This structure contains LOC Inject Position parameters

#### Parameters

<i>pLatitude</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <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> <li>• Note - This field must be specified together with pLongitude and pHorUncCircular.</li> </ul>
<i>pLongitude</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <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> <li>• Note - This field must be specified together with pLatitude and pHorUncCircular.</li> </ul>
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> <li>• Note - This field must be specified together with pLatitude and pLongitude.</li> </ul>
<i>pHorConfidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal confidence.</li> <li>• Units - Percent</li> <li>• Values <ul style="list-style-type: none"> <li>– Valid Values - 1 to 99</li> <li>– Invalid Values - 0, 101 to 255</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with horizontal uncertainty. If not specified when pHorUncCircular is set, the default value is 50.</li> </ul>
<i>pHorReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>

- Values
  - 0 - Location reliability is not set.
  - 1 - Location reliability is very low; use it at your own risk
  - 2 - Location reliability is low; little or no cross-checking is possible.
  - 3 - Location reliability is medium; limited cross-check passed
  - 4 - Location reliability is high; strong cross-check passed

## Parameters

<i>pAltitudeWrtEllipsoid</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Altitude With Respect to Ellipsoid.</li> <li>• Units - Meters</li> <li>• Values <ul style="list-style-type: none"> <li>– Positive - height</li> <li>– Negative = depth</li> </ul> </li> </ul>
<i>pAltitudeWrtMeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Altitude With Respect to Sea Level.</li> <li>• Units - Meters</li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> <li>• Note - This is mandatory if either <i>pAltitudeWrtEllipsoid</i> or <i>pAltitudeWrtMeanSeaLevel</i> is specified.</li> </ul>
<i>pVertConfidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Vertical confidence.</li> <li>• Units - Percentage</li> <li>• Values <ul style="list-style-type: none"> <li>– Valid Values - 0 to 99</li> <li>– Invalid Values - 0, 100-256</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with the vertical uncertainty. If not specified, the default value will be 50.</li> </ul>
<i>pVertReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>

- Values
  - 0 - Location reliability is not set.
  - 1 - Location reliability is very low; use it at your own risk.
  - 2 - Location reliability is low; little or no cross-checking is possible
  - 3 - Location reliability is medium; limited cross-check passed
  - 4 - Location reliability is high; strong cross-check passed

## Parameters

<i>pAltitudeSrcInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> </ul>
-------------------------	--

- Pointer to struct [altitudeSrcInfo](#). See [altitudeSrcInfo](#) for more information

## Parameters

<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> </ul>
<i>pTimestampAge</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Position age, which is an estimate of how long ago this fix was made.</li> <li>• Units - Milliseconds</li> </ul>
<i>pPositionSrc</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Source from which this position was obtained</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Position source is GNSS</li> <li>– 1 - Position source is Cell ID</li> <li>– 2 - Position source is Enhanced Cell ID</li> <li>– 3 - Position source is Wi-Fi</li> <li>– 4 - Position source is Terrestrial</li> <li>– 5 - Position source is GNSS Terrestrial Hybrid</li> <li>– 6 - Other sources</li> </ul> </li> <li>• Note - If altitude is specified and the altitude source is not specified, the engine assumes that the altitude was obtained using the specified position source. <ul style="list-style-type: none"> <li>– If both altitude and altitude source are specified, the engine assumes that only latitude and longitude were obtained using the specified position source.</li> </ul> </li> </ul>
<i>pRawHorUnc-Circular</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal position uncertainty (circular) without any optimization.</li> <li>• Units - Meters</li> </ul>
<i>pRawHor-Confidence</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Horizontal confidence associated with raw horizontal uncertainty</li> <li>• Units: Percent</li> <li>• Values <ul style="list-style-type: none"> <li>– Valid values - 1 to 99</li> <li>– Invalid values - 0, 101 to 255</li> <li>– If 100 is received, reinterpret to 99</li> </ul> </li> <li>• Note - This field must be specified together with raw horizontal uncertainty. If not specified when rawHorUncCircular is set, the default value is 50.</li> </ul>

## 8.299.2 Field Documentation

8.299.2.1 altitudeSrcInfo\* LocInjectPositionReq::pAltitudeSrcInfo

8.299.2.2 ULONG\* LocInjectPositionReq::pAltitudeWrtEllipsoid

8.299.2.3 ULONG\* LocInjectPositionReq::pAltitudeWrtMeanSeaLevel

- 8.299.2.4 **BYTE\*** `LocInjectPositionReq::pHorConfidence`
- 8.299.2.5 **ULONG\*** `LocInjectPositionReq::pHorReliability`
- 8.299.2.6 **ULONG\*** `LocInjectPositionReq::pHorUncCircular`
- 8.299.2.7 **ULONGLONG\*** `LocInjectPositionReq::pLatitude`
- 8.299.2.8 **ULONGLONG\*** `LocInjectPositionReq::pLongitude`
- 8.299.2.9 **ULONG\*** `LocInjectPositionReq::pPositionSrc`
- 8.299.2.10 **BYTE\*** `LocInjectPositionReq::pRawHorConfidence`
- 8.299.2.11 **ULONG\*** `LocInjectPositionReq::pRawHorUncCircular`
- 8.299.2.12 **ULONG\*** `LocInjectPositionReq::pTimestampAge`
- 8.299.2.13 **ULONGLONG\*** `LocInjectPositionReq::pTimestampUtc`
- 8.299.2.14 **BYTE\*** `LocInjectPositionReq::pVertConfidence`
- 8.299.2.15 **ULONG\*** `LocInjectPositionReq::pVertReliability`
- 8.299.2.16 **ULONG\*** `LocInjectPositionReq::pVertUnc`

## 8.300 LocInjectSensorDataReq Struct Reference

### Data Fields

- [ULONG](#) \* `pOpaqueIdentifier`
- `sensorData` \* [pAcceleroData](#)
- `sensorData` \* [pGyroData](#)
- [ULONG](#) \* `pAcceleroTimeSrc`
- [ULONG](#) \* `pGyroTimeSrc`
- `tempratureData` \* [pAcceleroTempData](#)
- `tempratureData` \* [pGyroTempData](#)

### 8.300.1 Detailed Description

This structure contains parameters to inject sensor data into the GNSS location engine

#### Parameters

<i>pOpaque- Identifier</i>	<ul style="list-style-type: none"> <li>• Opaque Identifier (Optional parameter)</li> <li>• An opaque identifier that is sent in by the client that will be echoed in the indication so the client can relate the indication to the request.</li> </ul>
<i>pAcceleroData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">sensorData</a>. See <a href="#">sensorData</a> for more information</li> </ul>

<i>pGyroData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">sensorData</a>. See <a href="#">sensorData</a> for more information</li> </ul>
<i>pAcceleroTimeSrc</i>	<ul style="list-style-type: none"> <li>• 3-Axis Accelerometer Data Time Source (Optional parameter)</li> <li>• The location service uses this field to identify the time reference used in the accelerometer data time stamps.</li> <li>• If not specified, the location service assumes that the time source for the accelerometer data is unknown.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>pGyroTimeSrc</i>	<ul style="list-style-type: none"> <li>• 3-Axis Gyroscope Data Time Source (Optional)</li> <li>• The location service uses this field to identify the time reference used in the gyroscope data time stamps.</li> <li>• If not specified, the location service assumes that the time source for the gyroscope data is unknown.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>pAcceleroTempData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">tempratureData</a>. See <a href="#">tempratureData</a> for more information</li> </ul>
<i>pGyroTempData</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Pointer to struct <a href="#">tempratureData</a>. See <a href="#">tempratureData</a> for more information</li> </ul>

## 8.300.2 Field Documentation

8.300.2.1 **sensorData\*** LocInjectSensorDataReq::pAcceleroData

8.300.2.2 **tempratureData\*** LocInjectSensorDataReq::pAcceleroTempData

8.300.2.3 **ULONG\*** LocInjectSensorDataReq::pAcceleroTimeSrc

8.300.2.4 **sensorData\*** LocInjectSensorDataReq::pGyroData

8.300.2.5 **tempratureData\*** LocInjectSensorDataReq::pGyroTempData

8.300.2.6 **ULONG\*** LocInjectSensorDataReq::pGyroTimeSrc

8.300.2.7 **ULONG\*** LocInjectSensorDataReq::pOpaqueldentifier

## 8.301 LocSetCradleMountReq Struct Reference

## Data Fields

- [ULONG state](#)
- [BYTE \\* pConfidence](#)

### 8.301.1 Detailed Description

This structure contains parameters to set current cradle mount configuration

#### Parameters

<i>state</i>	<ul style="list-style-type: none"> <li>• Cradle Mount State</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 0 - Device is mounted on the cradle</li> <li>– 1 - Device is not mounted on the cradle</li> <li>– 2 - Unknown cradle mount state</li> </ul> </li> </ul>
<i>pConfidence</i>	<ul style="list-style-type: none"> <li>• Cradle Mount Confidence (Optional)</li> <li>• Confidence in the Cradle Mount state expressed as a percentage.</li> <li>• Range - 0 to 100</li> </ul>

### 8.301.2 Field Documentation

8.301.2.1 **BYTE\*** `LocSetCradleMountReq::pConfidence`

8.301.2.2 **ULONG** `LocSetCradleMountReq::state`

## 8.302 LOCStartReq Struct Reference

## Data Fields

- [BYTE SessionId](#)
- [ULONG \\* pRecurrenceType](#)
- [ULONG \\* pHorizontalAccuracyLvl](#)
- [ULONG \\* pIntermediateReportState](#)
- [ULONG \\* pMinIntervalTime](#)
- [struct LocApplicationInfo \\* pApplicationInfo](#)
- [ULONG \\* pConfigAltitudeAssumed](#)

### 8.302.1 Detailed Description

This structure contains the LOC Start Request

#### Parameters

<i>SessionId[IN]</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> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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-AccuracyLv</i> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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>pMinInterval-Time</i> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <li>• Minimum time interval, specified by the control point, that must elapse between position reports.</li> <li>• Units - Milliseconds</li> <li>• Default - 1000 ms</li> </ul>
<i>LocApplication-Info</i> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <li>• LOC Application Parameters</li> <li>• See <a href="#">LocApplicationInfo</a> for more information</li> </ul>
<i>pConfigAltitude-Assumed</i> [IN]	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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>

## 8.302.2 Field Documentation

8.302.2.1 struct LocApplicationInfo\* LOCStartReq::pApplicationInfo

8.302.2.2 ULONG\* LOCStartReq::pConfigAltitudeAssumed

8.302.2.3 **ULONG\*** LOCStartReq::pHorizontalAccuracyLvl

8.302.2.4 **ULONG\*** LOCStartReq::pIntermediateReportState

8.302.2.5 **ULONG\*** LOCStartReq::pMinIntervalTime

8.302.2.6 **ULONG\*** LOCStartReq::pRecurrenceType

8.302.2.7 **BYTE** LOCStartReq::SessionId

## 8.303 LOCStopReq Struct Reference

### Data Fields

- [BYTE sessionId](#)

### 8.303.1 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>
------------------	--

### 8.303.2 Field Documentation

8.303.2.1 **BYTE** LOCStopReq::sessionId

## 8.304 LteCQIParm Struct Reference

### Data Fields

- [BYTE ValidityCW0](#)
- [BYTE CQIValueCW0](#)
- [BYTE ValidityCW1](#)
- [BYTE CQIValueCW1](#)

### 8.304.1 Detailed Description

This structure contains information about the SLQSSwiGetLteCQI response parameters.

#### Parameters

<i>pValidityCW0[O-UT]</i>	<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[OUT]</i>	<ul style="list-style-type: none"> <li>Values           <ul style="list-style-type: none"> <li>Range 0~15</li> </ul> </li> </ul>
<i>pValidityCW1[OUT]</i>	<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[OUT]</i>	<ul style="list-style-type: none"> <li>Values           <ul style="list-style-type: none"> <li>Range 0~15</li> </ul> </li> </ul>

### 8.304.2 Field Documentation

8.304.2.1 BYTE LteCQIParm::CQIValueCW0

8.304.2.2 BYTE LteCQIParm::CQIValueCW1

8.304.2.3 BYTE LteCQIParm::ValidityCW0

8.304.2.4 BYTE LteCQIParm::ValidityCW1

## 8.305 LteEARFCN Struct Reference

### Data Fields

- [BYTE status](#)
- [ULONG earfcn0](#)
- [ULONG earfcn1](#)

### 8.305.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>0 - Disable</li> <li>1 - Enable</li> </ul>
<i>earfcn0</i>	<ul style="list-style-type: none"> <li>Primary DL EARFCN to which the UE is locked</li> </ul>
<i>earfcn1</i>	<ul style="list-style-type: none"> <li>Secondary DL EARFCN to which the UE is locked</li> <li>Note : Make earfcn1 value equal to earfcn0 if only one EARFCN is desired.</li> </ul>

### 8.305.2 Field Documentation

8.305.2.1 **ULONG** `IteEARFCN::earfcn0`

8.305.2.2 **ULONG** `IteEARFCN::earfcn1`

8.305.2.3 **BYTE** `IteEARFCN::status`

## 8.306 `IteGsmCellInfo` Struct Reference

### Data Fields

- [BYTE](#) `cellReselPriority`
- [BYTE](#) `threshGsmHigh`
- [BYTE](#) `threshGsmLow`
- [BYTE](#) `nccPermitted`
- [BYTE](#) `cells_len`
- [gsmCellInfo](#) `GsmCellInfo` [255]

### 8.306.1 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>

## 8.306.2 Field Documentation

8.306.2.1 **BYTE** `IteGsmCellInfo::cellReselPriority`

8.306.2.2 **BYTE** `IteGsmCellInfo::cells_len`

8.306.2.3 **gsmCellInfo** `IteGsmCellInfo::GsmCellInfo[255]`

8.306.2.4 **BYTE** `IteGsmCellInfo::nccPermitted`

8.306.2.5 **BYTE** `IteGsmCellInfo::threshGsmHigh`

8.306.2.6 **BYTE** `IteGsmCellInfo::threshGsmLow`

## 8.307 LTEInfo Struct Reference

### Data Fields

- [BYTE](#) `band`
- [BYTE](#) `bandwidth`
- [WORD](#) `RXChan`
- [WORD](#) `TXChan`
- [BYTE](#) `emmState`
- [BYTE](#) `emmSubState`
- [BYTE](#) `emmConnState`

### 8.307.1 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 ~ 41 (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> <li>– 0x07 - IMSI Detach</li> <li>– 0x08 - Waiting for ESM</li> </ul> </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>

### 8.307.2 Field Documentation

8.307.2.1 BYTE LTEInfo::band

8.307.2.2 BYTE LTEInfo::bandwidth

8.307.2.3 BYTE LTEInfo::emmConnState

8.307.2.4 BYTE LTEInfo::emmState

8.307.2.5 BYTE LTEInfo::emmSubState

8.307.2.6 WORD LTEInfo::RXChan

8.307.2.7 WORD LTEInfo::TXChan

## 8.308 LTEInfoInterfreq Struct Reference

### Data Fields

- [BYTE ueInIdle](#)
- [BYTE freqsLen](#)
- [infoInterFreq InfoInterFreq](#) [255]

### 8.308.1 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[MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">infoInterFreq</a> for more information.</li> </ul>

### 8.308.2 Field Documentation

8.308.2.1 BYTE LTEInfoInterfreq::freqsLen

8.308.2.2 [infoInterFreq](#) LTEInfoInterfreq::InfoInterFreq[255]

8.308.2.3 BYTE LTEInfoInterfreq::ueInIdle

## 8.309 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 \[255\]](#)

### 8.309.1 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. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </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 ue_in_idle 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 ue_in_idle is TRUE. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>threshServingLow</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>

## 8.309.2 Field Documentation

8.309.2.1 **cellParams** LTEInfoIntrafreq::CellParams[255]

8.309.2.2 **BYTE** LTEInfoIntrafreq::cellReselPriority

8.309.2.3 **BYTE** LTEInfoIntrafreq::cellsLen

8.309.2.4 **WORD** LTEInfoIntrafreq::earfcn

8.309.2.5 **ULONG** LTEInfoIntrafreq::globalCellId

8.309.2.6 **BYTE** LTEInfoIntrafreq::plmn[3]

8.309.2.7 **WORD** LTEInfoIntrafreq::servingCellId

8.309.2.8 **BYTE** LTEInfoIntrafreq::sIntraSearch

8.309.2.9 **BYTE** LTEInfoIntrafreq::sNonIntraSearch

8.309.2.10 **WORD** LTEInfoIntrafreq::tac

8.309.2.11 **BYTE** LTEInfoIntrafreq::threshServingLow

8.309.2.12 **BYTE** LTEInfoIntrafreq::ueIdle

## 8.310 LTEInfoNeighboringGSM Struct Reference

### Data Fields

- [BYTE ueIdle](#)
- [BYTE freqsLen](#)
- [lteGsmCellInfo](#) [LteGsmCellInfo](#) [255]

### 8.310.1 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. <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 GSM cell information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>LteGsmCellInfo</i> [ MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• See <a href="#">lteGsmCellInfo</a> for more information.</li> </ul>

### 8.310.2 Field Documentation

8.310.2.1 **BYTE** LTEInfoNeighboringGSM::freqsLen

8.310.2.2 **lteGsmCellInfo** LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.310.2.3 **BYTE** LTEInfoNeighboringGSM::ueIdle

## 8.311 LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

- [BYTE ueIdle](#)
- [BYTE freqsLen](#)
- [lteWcdmaCellInfo](#) [LTEWCDMACellInfo](#) [255]

### 8.311.1 Detailed Description

This structure contains information about the LTE Neighboring WCDMA 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 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>

## 8.311.2 Field Documentation

8.311.2.1 BYTE LTEInfoNeighboringWCDMA::freqsLen

8.311.2.2 lteWcdmaCellInfo LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]

8.311.2.3 BYTE LTEInfoNeighboringWCDMA::ueInIdle

## 8.312 LteNasReleaseInfo\_s Struct Reference

## Data Fields

- [BYTE nas\\_release](#)
- [BYTE nas\\_major](#)
- [BYTE nas\\_minor](#)

## 8.312.1 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>

## 8.312.2 Field Documentation

8.312.2.1 BYTE LteNasReleaseInfo\_s::nas\_major

8.312.2.2 BYTE LteNasReleaseInfo\_s::nas\_minor

8.312.2.3 **BYTE** `LteNasReleaseInfo_s::nas_release`

## 8.313 `ItePCI` Struct Reference

### Data Fields

- [BYTE](#) `status`
- [ULONG](#) `earfcn`
- [ULONG](#) `pci`

### 8.313.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• 0 - Disable</li> <li>• 1 - Enable</li> </ul>
<i>earfcn</i>	<ul style="list-style-type: none"> <li>• UARFCN to which UE is locked</li> </ul>
<i>pci</i>	<ul style="list-style-type: none"> <li>• PCI to which the UE is locked</li> </ul>

### 8.313.2 Field Documentation

8.313.2.1 **ULONG** `ItePCI::earfcn`

8.313.2.2 **ULONG** `ItePCI::pci`

8.313.2.3 **BYTE** `ItePCI::status`

## 8.314 `IteRsrpinformation` Struct Reference

### Data Fields

- [SHORT](#) `rsrplevel`

### 8.314.1 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>
------------------	--

### 8.314.2 Field Documentation

8.314.2.1 `SHORT lteRsrpInformation::rsrplevel`

## 8.315 LTERSRPThresh Struct Reference

### Data Fields

- [BYTE LTERSRPThreshListLen](#)
- [WORD \\* pLTERSRPThreshList](#)

### 8.315.1 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>

### 8.315.2 Field Documentation

8.315.2.1 `BYTE LTERSRPThresh::LTERSRPThreshListLen`

8.315.2.2 `WORD* LTERSRPThresh::pLTERSRPThreshList`

## 8.316 LTERSRQThresh Struct Reference

### Data Fields

- [BYTE LTERSRQThreshListLen](#)
- [WORD \\* pLTERSRQThreshList](#)

### 8.316.1 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>

### 8.316.2 Field Documentation

8.316.2.1 **BYTE** LTERSRQThresh::LTERSRQThreshListLen

8.316.2.2 **WORD\*** LTERSRQThresh::pLTERSRQThreshList

## 8.317 LTERSSIThresh Struct Reference

### Data Fields

- [BYTE](#) LTERSSIThreshListLen
- [WORD](#) \* [pLTERSSIThreshList](#)

### 8.317.1 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>

### 8.317.2 Field Documentation

8.317.2.1 **BYTE** LTERSSIThresh::LTERSSIThreshListLen

8.317.2.2 **WORD\*** LTERSSIThresh::pLTERSSIThreshList

## 8.318 LTESigRptCfg Struct Reference

### Data Fields

- [BYTE](#) rptRate
- [BYTE](#) avgPeriod

### 8.318.1 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>

## 8.318.2 Field Documentation

8.318.2.1 BYTE LTESigRptCfg::avgPeriod

8.318.2.2 BYTE LTESigRptCfg::rptRate

## 8.319 LTESigRptConfig Struct Reference

## Data Fields

- [BYTE rptRate](#)
- [BYTE avgPeriod](#)

## 8.319.1 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>

## 8.319.2 Field Documentation

8.319.2.1 BYTE LTESigRptConfig::avgPeriod

8.319.2.2 BYTE LTESigRptConfig::rptRate

## 8.320 lteSnrinformation Struct Reference

## Data Fields

- [SHORT snrlevel](#)

## 8.320.1 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>
-----------------	---

## 8.320.2 Field Documentation

8.320.2.1 SHORT lteSnrinformation::snrlevel

## 8.321 LTESNRThresh Struct Reference

### Data Fields

- [BYTE LTESNRThresListLen](#)
- [SHORT \\* pLTESNRThresList](#)

### 8.321.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

#### Parameters

<i>LTESNRThres- ListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
<i>pLTESNRThres- List</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>

### 8.321.2 Field Documentation

8.321.2.1 BYTE LTESNRThresh::LTESNRThresListLen

8.321.2.2 SHORT\* LTESNRThresh::pLTESNRThresList

## 8.322 LTESNRThreshold Struct Reference

### Data Fields

- [BYTE LTESNRThreshListLen](#)
- [WORD \\* pLTESNRThreshList](#)

### 8.322.1 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>
--------------------------------	--

### 8.322.2 Field Documentation

8.322.2.1 **BYTE** LTESNRThreshold::LTESNRThreshListLen

8.322.2.2 **WORD\*** LTESNRThreshold::pLTESNRThreshList

## 8.323 LTESSInfo Struct Reference

### Data Fields

- [INT8](#) rssi
- [INT8](#) rsrq
- [SHORT](#) rsrp
- [SHORT](#) snr

### 8.323.1 Detailed Description

This structure contains the parameters for LTE 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. <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>

### 8.323.2 Field Documentation

8.323.2.1 **SHORT** LTESSInfo::rsrp

8.323.2.2 **INT8** LTESSInfo::rsrq

8.323.2.3 INT8 LTESSInfo::rssi

8.323.2.4 SHORT LTESSInfo::snr

## 8.324 lteSSInfo Struct Reference

### Data Fields

- int8\_t [rssi](#)
- int8\_t [rsrq](#)
- int16\_t [rsrp](#)
- int16\_t [snr](#)

### 8.324.1 Detailed Description

#### Parameters

<i>rssi</i>	RSSI in dBm.
<i>rsrq</i>	RSRQ value in dB
<i>rsrp</i>	Current RSRP in dBm as measured by L1.
<i>snr</i>	SNR level as a scaled integer in units of 0.1 dB.

### 8.324.2 Field Documentation

8.324.2.1 int16\_t lteSSInfo::rsrp

8.324.2.2 int8\_t lteSSInfo::rsrq

8.324.2.3 int8\_t lteSSInfo::rssi

8.324.2.4 int16\_t lteSSInfo::snr

## 8.325 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](#)

### 8.325.1 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>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>
<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> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 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>

### 8.325.2 Field Documentation

8.325.2.1 **ULONG** lteSysInfo::cellId

8.325.2.2 **BYTE** lteSysInfo::cellIdValid

8.325.2.3 **WORD** lteSysInfo::lac

8.325.2.4 **BYTE** lteSysInfo::lacValid

8.325.2.5 **BYTE** lteSysInfo::MCC[3]

8.325.2.6 **BYTE** lteSysInfo::MNC[3]

8.325.2.7 **BYTE** lteSysInfo::networkIdValid

8.325.2.8 **BYTE** lteSysInfo::regRejectInfoValid

8.325.2.9 **BYTE** lteSysInfo::rejCause

8.325.2.10 **BYTE** lteSysInfo::rejectSrvDomain

8.325.2.11 **sysInfoCommon** lteSysInfo::sysInfoLTE

8.325.2.12 **WORD** lteSysInfo::tac

8.325.2.13 **BYTE** lteSysInfo::tacValid

## 8.326 lteWcdmaCellInfo Struct Reference

## Data Fields

- [WORD](#) `uarfcn`
- [BYTE](#) `cellReselPriority`
- [WORD](#) `threshXhigh`
- [WORD](#) `threshXlow`
- [BYTE](#) `cellsLen`
- [wcdmaCellInfo](#) [WCDMACellInfo](#) [255]

### 8.326.1 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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>WCDMACell-Info[<code>MAX_DESCRIPTION_LENGTH</code>]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">wcdmaCellInfo</a> for more information.</li> </ul>

### 8.326.2 Field Documentation

8.326.2.1 **BYTE** `lteWcdmaCellInfo::cellReselPriority`

8.326.2.2 **BYTE** `lteWcdmaCellInfo::cellsLen`

8.326.2.3 **WORD** `lteWcdmaCellInfo::threshXhigh`

8.326.2.4 **WORD** `lteWcdmaCellInfo::threshXlow`

8.326.2.5 **WORD** `lteWcdmaCellInfo::uarfcn`

8.326.2.6 wcdmaCellInfo lteWcdmaCellInfo::WCDMACellInfo[255]

## 8.327 messageModeTlv Struct Reference

### Data Fields

- [uint8\\_t TlvPresent](#)
- [sMSMessageModelInfo MessageModelInfo](#)

### 8.327.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Boolean indicating the presence of the TLV in the QMI response</li></ul>
<i>MessageMode-Info</i>	<ul style="list-style-type: none"><li>• Message Mode</li><li>• See <a href="#">sMSMessageModelInfo</a> for more information</li></ul>

### 8.327.2 Field Documentation

8.327.2.1 [sMSMessageModelInfo](#) messageModeTlv::MessageModelInfo

8.327.2.2 [uint8\\_t](#) messageModeTlv::TlvPresent

## 8.328 messageWaitingInfoContent Struct Reference

### Data Fields

- [BYTE msgType](#)
- [BYTE activeInd](#)
- [BYTE msgCount](#)

### 8.328.1 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>

### 8.328.2 Field Documentation

8.328.2.1 BYTE messageWaitingInfoContent::activeInd

8.328.2.2 BYTE messageWaitingInfoContent::msgCount

8.328.2.3 BYTE messageWaitingInfoContent::msgType

## 8.329 minBasedIMSI Struct Reference

### Data Fields

- [BYTE mccM](#) [3]
- [WORD imsiM1112](#)
- [BYTE imsiMS1](#) [7]
- [BYTE imsiMS2](#) [3]

### 8.329.1 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>

### 8.329.2 Field Documentation

8.329.2.1 WORD minBasedIMSI::imsiM1112

8.329.2.2 BYTE minBasedIMSI::imsiMS1[7]

8.329.2.3 **BYTE** minBasedIMSI::imsiMS2[3]

8.329.2.4 **BYTE** minBasedIMSI::mccM[3]

## 8.330 mitigationDevList Struct Reference

### Data Fields

- [BYTE](#) mitigationDevIdLen
- [CHAR](#) mitigationDevId [255]
- [BYTE](#) maxMitigationLevel

### 8.330.1 Detailed Description

This structure contains mitigation devices list

#### Parameters

<i>mitigationDevId- Len</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements<ul style="list-style-type: none"><li>– mitigation_dev_id</li></ul></li></ul>
<i>mitigationDevId</i>	<ul style="list-style-type: none"><li>• Mitigation device ID.</li></ul>
<i>maxMitigation- Level</i>	<ul style="list-style-type: none"><li>• Maximum valid mitigation level.</li><li>• Valid range - 0 to max_mitigation_level.</li></ul>

### 8.330.2 Field Documentation

8.330.2.1 **BYTE** mitigationDevList::maxMitigationLevel

8.330.2.2 **CHAR** mitigationDevList::mitigationDevId[255]

8.330.2.3 **BYTE** mitigationDevList::mitigationDevIdLen

## 8.331 MNRInfo Struct Reference

### Data Fields

- [WORD](#) mcc
- [WORD](#) mnc
- [ULONG](#) rat

### 8.331.1 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>

## 8.331.2 Field Documentation

8.331.2.1 WORD MNRInfo::mcc

8.331.2.2 WORD MNRInfo::mnc

8.331.2.3 ULONG MNRInfo::rat

## 8.332 ModifyProfileIn Struct Reference

## Data Fields

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

## 8.332.1 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 (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>• Contains Union of profile(3GPP/3GPP2) structures</li> </ul>

### 8.332.2 Field Documentation

#### 8.332.2.1 QmiProfileInfo ModifyProfileIn::curProfile

#### 8.332.2.2 BYTE\* ModifyProfileIn::pProfileID

#### 8.332.2.3 BYTE\* ModifyProfileIn::pProfileType

## 8.333 ModifyProfileOut Struct Reference

### Data Fields

- [USHORT](#) \* [pExtErrorCode](#)

### 8.333.1 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 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>
----------------------	--

### 8.333.2 Field Documentation

#### 8.333.2.1 USHORT\* ModifyProfileOut::pExtErrorCode

## 8.334 msgWaitingInfo Struct Reference

### Data Fields

- [BYTE](#) numInstances
- [messageWaitingInfoContent](#) msgWaitInfo [0xFF]

### 8.334.1 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>

### 8.334.2 Field Documentation

8.334.2.1 `messageWaitingInfoContent` `msgWaitingInfo::msgWaitInfo[0xFF]`

8.334.2.2 `BYTE` `msgWaitingInfo::numInstances`

## 8.335 `namName` Struct Reference

### Data Fields

- `BYTE` `namNameLen`
- `BYTE` `namName` [12]

### 8.335.1 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>namName</i>	<ul style="list-style-type: none"> <li>• Name information in ASCII. The maximum length of <code>nam_name</code> is 12.</li> </ul>

### 8.335.2 Field Documentation

8.335.2.1 `BYTE` `namName::namName[12]`

8.335.2.2 `BYTE` `namName::namNameLen`

## 8.336 `nas_acqOrderPref` Struct Reference

### Data Fields

- `uint8_t` `acqOrdeLen`
- `uint8_t *` `pAcqOrder`

### 8.336.1 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_1xEVDO - 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>

## 8.336.2 Field Documentation

8.336.2.1 uint8\_t nas\_acqOrderPref::acqOrdeLen

8.336.2.2 uint8\_t\* nas\_acqOrderPref::pAcqOrder

## 8.337 nas\_AddCDMASysInfo Struct Reference

## Data Fields

- uint16\_t [geoSysIdx](#)
- uint16\_t [regPrd](#)

## 8.337.1 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>

## 8.337.2 Field Documentation

8.337.2.1 uint16\_t nas\_AddCDMASysInfo::geoSysIdx

8.337.2.2 uint16\_t nas\_AddCDMASysInfo::regPrd

## 8.338 nas\_AddSysInfo Struct Reference

## Data Fields

- uint16\_t [geoSysIdx](#)
- uint32\_t [cellBroadcastCap](#)

### 8.338.1 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>

### 8.338.2 Field Documentation

8.338.2.1 uint32\_t nas\_AddSysInfo::cellBroadcastCap

8.338.2.2 uint16\_t nas\_AddSysInfo::geoSysIdx

## 8.339 nas\_CallBarringSysInfo Struct Reference

## Data Fields

- uint32\_t [csBarStatus](#)
- uint32\_t [psBarStatus](#)

### 8.339.1 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>
--------------------	---

### 8.339.2 Field Documentation

8.339.2.1 uint32\_t nas\_CallBarringSysInfo::csBarStatus

8.339.2.2 uint32\_t nas\_CallBarringSysInfo::psBarStatus

## 8.340 nas\_callBarStatus Struct Reference

### Data Fields

- uint32\_t [csBarStatus](#)
- uint32\_t [psBarStatus](#)

### 8.340.1 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>

### 8.340.2 Field Documentation

8.340.2.1 `uint32_t nas_callBarStatus::csBarStatus`

8.340.2.2 `uint32_t nas_callBarStatus::psBarStatus`

## 8.341 nas\_CDMAECIOThresh Struct Reference

### Data Fields

- `uint8_t` [CDMAECIOThreshListLen](#)
- `int16_t *` [pCDMAECIOThreshList](#)

### 8.341.1 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>

### 8.341.2 Field Documentation

8.341.2.1 `uint8_t nas_CDMAECIOThresh::CDMAECIOThreshListLen`

8.341.2.2 `int16_t*` `nas_CDMAECIOThresh::pCDMAECIOThreshList`

## 8.342 nas\_CDMAInfo Struct Reference

### Data Fields

- `uint16_t` [sid](#)
- `uint16_t` [nid](#)
- `uint16_t` [baseId](#)
- `uint16_t` [refpn](#)
- `uint32_t` [baseLat](#)
- `uint32_t` [baseLong](#)

### 8.342.1 Detailed Description

This structure contains information about the CDMA Network.

#### Parameters

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

## 8.342.2 Field Documentation

8.342.2.1 uint16\_t nas\_CDMAInfo::baseId

8.342.2.2 uint32\_t nas\_CDMAInfo::baseLat

8.342.2.3 uint32\_t nas\_CDMAInfo::baseLong

8.342.2.4 uint16\_t nas\_CDMAInfo::nid

8.342.2.5 uint16\_t nas\_CDMAInfo::refpn

8.342.2.6 uint16\_t nas\_CDMAInfo::sid

## 8.343 nas\_CDMARSSIThresh Struct Reference

### Data Fields

- uint8\_t [CDMARSSIThreshListLen](#)
- int16\_t \* [pCDMARSSIThreshList](#)

### 8.343.1 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>
----------------------------------	--

### 8.343.2 Field Documentation

8.343.2.1 `uint8_t nas_CDMARSSIThresh::CDMARSSIThreshListLen`

8.343.2.2 `int16_t* nas_CDMARSSIThresh::pCDMARSSIThreshList`

## 8.344 nas\_CDMA SysInfo Struct Reference

### Data Fields

- [nas\\_sysInfoCommon](#) `sysInfoCDMA`
- `uint8_t isSysPrIMatchValid`
- `uint8_t isSysPrIMatch`
- `uint8_t pRevInUseValid`
- `uint8_t pRevInUse`
- `uint8_t bsPRevValid`
- `uint8_t bsPRev`
- `uint8_t ccsSupportedValid`
- `uint8_t ccsSupported`
- `uint8_t cdmaSysIdValid`
- `uint16_t systemID`
- `uint16_t networkID`
- `uint8_t bsInfoValid`
- `uint16_t baseId`
- `uint32_t baseLat`
- `uint32_t baseLong`
- `uint8_t packetZoneValid`
- `uint16_t packetZone`
- `uint8_t networkIdValid`
- `uint8_t MCC` [3]
- `uint8_t MNC` [3]

### 8.344.1 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>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>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> <li>• MCC wildcard value is returned as {'3', 0xFF, 0xFF}.</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> <li>• MNC wildcard value is returned as {'7', 0xFF, 0xFF}.</li> </ul>
-------------------------	--

## 8.344.2 Field Documentation

- 8.344.2.1 uint16\_t nas\_CDMASysInfo::baseId
- 8.344.2.2 uint32\_t nas\_CDMASysInfo::baseLat
- 8.344.2.3 uint32\_t nas\_CDMASysInfo::baseLong
- 8.344.2.4 uint8\_t nas\_CDMASysInfo::bsInfoValid
- 8.344.2.5 uint8\_t nas\_CDMASysInfo::bsPRev
- 8.344.2.6 uint8\_t nas\_CDMASysInfo::bsPRevValid
- 8.344.2.7 uint8\_t nas\_CDMASysInfo::ccsSupported
- 8.344.2.8 uint8\_t nas\_CDMASysInfo::ccsSupportedValid
- 8.344.2.9 uint8\_t nas\_CDMASysInfo::cdmaSysIdValid
- 8.344.2.10 uint8\_t nas\_CDMASysInfo::isSysPriMatch
- 8.344.2.11 uint8\_t nas\_CDMASysInfo::isSysPriMatchValid
- 8.344.2.12 uint8\_t nas\_CDMASysInfo::MCC[3]
- 8.344.2.13 uint8\_t nas\_CDMASysInfo::MNC[3]
- 8.344.2.14 uint16\_t nas\_CDMASysInfo::networkId
- 8.344.2.15 uint8\_t nas\_CDMASysInfo::networkIdValid
- 8.344.2.16 uint16\_t nas\_CDMASysInfo::packetZone
- 8.344.2.17 uint8\_t nas\_CDMASysInfo::packetZoneValid
- 8.344.2.18 uint8\_t nas\_CDMASysInfo::pRevInUse
- 8.344.2.19 uint8\_t nas\_CDMASysInfo::pRevInUseValid
- 8.344.2.20 nas\_sysInfoCommon nas\_CDMASysInfo::sysInfoCDMA
- 8.344.2.21 uint16\_t nas\_CDMASysInfo::systemId

## 8.345 nas\_CDMASysInfoExt Struct Reference

## Data Fields

- uint16\_t [MCC](#)
- uint8\_t [imsi\\_11\\_12](#)

### 8.345.1 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>

### 8.345.2 Field Documentation

8.345.2.1 uint8\_t nas\_CDMA SysInfoExt::imsi\_11\_12

8.345.2.2 uint16\_t nas\_CDMA SysInfoExt::MCC

## 8.346 nas\_cellParams Struct Reference

## Data Fields

- uint16\_t [pci](#)
- int16\_t [rsrq](#)
- int16\_t [rsrp](#)
- int16\_t [rssi](#)
- int16\_t [srxlev](#)

### 8.346.1 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>

## 8.346.2 Field Documentation

8.346.2.1 uint16\_t nas\_cellParams::pci

8.346.2.2 int16\_t nas\_cellParams::rsrp

8.346.2.3 int16\_t nas\_cellParams::rsrq

8.346.2.4 int16\_t nas\_cellParams::rssI

8.346.2.5 int16\_t nas\_cellParams::srxlev

## 8.347 nas\_CommlInfo Struct Reference

### Data Fields

- int8\_t [temperature](#)
- uint8\_t [modemMode](#)
- uint8\_t [systemMode](#)
- uint8\_t [imsRegState](#)
- uint8\_t [psState](#)

### 8.347.1 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>

## 8.347.2 Field Documentation

8.347.2.1 uint8\_t nas\_CommlInfo::imsRegState

8.347.2.2 uint8\_t nas\_CommlInfo::modemMode

8.347.2.3 uint8\_t nas\_CommlInfo::psState

8.347.2.4 uint8\_t nas\_CommlInfo::systemMode

8.347.2.5 int8\_t nas\_CommlInfo::temperature

## 8.348 nas\_CSGID Struct Reference

### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- uint8\_t [mncPcsDigits](#)
- uint32\_t [id](#)
- uint8\_t [rat](#)

### 8.348.1 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>

## 8.348.2 Field Documentation

8.348.2.1 uint32\_t nas\_CSGID::id

8.348.2.2 uint16\_t nas\_CSGID::mcc

8.348.2.3 uint16\_t nas\_CSGID::mnc

8.348.2.4 uint8\_t nas\_CSGID::mncPcsDigits

8.348.2.5 uint8\_t nas\_CSGID::rat

## 8.349 nas\_currentPLMN Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [netDescrLength](#)
- uint8\_t [netDescr](#) [255]

### 8.349.1 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>

### 8.349.2 Field Documentation

8.349.2.1 uint16\_t nas\_currentPLMN::MCC

8.349.2.2 uint16\_t nas\_currentPLMN::MNC

8.349.2.3 uint8\_t nas\_currentPLMN::netDescr[255]

8.349.2.4 uint8\_t nas\_currentPLMN::netDescrLength

## 8.350 nas\_dataSrvCapabilities Struct Reference

### Data Fields

- uint8\_t [dataCapabilitiesLen](#)
- uint8\_t [dataCapabilities](#) [32]

### 8.350.1 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>

### 8.350.2 Field Documentation

8.350.2.1 uint8\_t nas\_dataSrvCapabilities::dataCapabilities[32]

8.350.2.2 uint8\_t nas\_dataSrvCapabilities::dataCapabilitiesLen

## 8.351 nas\_detailSvcInfo Struct Reference

## Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [srvCapability](#)
- uint8\_t [hdrSrvStatus](#)
- uint8\_t [hdrHybrid](#)
- uint8\_t [isSysForbidden](#)

### 8.351.1 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>
-----------------------	---

## 8.351.2 Field Documentation

8.351.2.1 `uint8_t nas_detailSvcInfo::hdrHybrid`

8.351.2.2 `uint8_t nas_detailSvcInfo::hdrSrvStatus`

8.351.2.3 `uint8_t nas_detailSvcInfo::isSysForbidden`

8.351.2.4 `uint8_t nas_detailSvcInfo::srvCapability`

8.351.2.5 `uint8_t nas_detailSvcInfo::srvStatus`

## 8.352 nas\_ecioListElement Struct Reference

### Data Fields

- `int16_t ecio`
- `uint8_t radiolf`

### 8.352.1 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_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>

## 8.352.2 Field Documentation

8.352.2.1 `int16_t nas_ecioListElement::ecio`

8.352.2.2 `uint8_t nas_ecioListElement::radiolf`

## 8.353 nas\_errorRateListElement Struct Reference

### Data Fields

- uint16\_t [errorRate](#)
- uint8\_t [radiolf](#)

### 8.353.1 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>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_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>

### 8.353.2 Field Documentation

8.353.2.1 uint16\_t nas\_errorRateListElement::errorRate

8.353.2.2 uint8\_t nas\_errorRateListElement::radiolf

## 8.354 nas\_GERANInfo Struct Reference

## Data Fields

- uint32\_t [cellID](#)
- uint8\_t [plmn](#) [3]
- uint16\_t [lac](#)
- uint16\_t [arfcn](#)
- uint8\_t [bsic](#)
- uint32\_t [timingAdvance](#)
- uint16\_t [rxLev](#)
- uint8\_t [nrmInst](#)
- [nas\\_nmrCellInfo](#) [insNmrCellInfo](#) [255]

## 8.354.1 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[NAS_PLM-N_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>nmlInst</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[ -NAS_MAX_DE- SCRIPTION_LE- NGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_nmrCellInfo</a> for more information.</li> </ul>

### 8.354.2 Field Documentation

8.354.2.1 `uint16_t nas_GERANInfo::arfcn`

8.354.2.2 `uint8_t nas_GERANInfo::bsic`

8.354.2.3 `uint32_t nas_GERANInfo::cellID`

8.354.2.4 `nas_nmrCellInfo nas_GERANInfo::insNmrCellInfo[255]`

8.354.2.5 `uint16_t nas_GERANInfo::lac`

8.354.2.6 `uint8_t nas_GERANInfo::nmlInst`

8.354.2.7 `uint8_t nas_GERANInfo::plmn[3]`

8.354.2.8 `uint16_t nas_GERANInfo::rxLev`

8.354.2.9 `uint32_t nas_GERANInfo::timingAdvance`

## 8.355 nas\_geranInstInfo Struct Reference

### Data Fields

- `uint16_t` [geranArfcn](#)
- `uint8_t` [geranBsicNcc](#)
- `uint8_t` [geranBsicBcc](#)
- `int16_t` [geranRssi](#)

### 8.355.1 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>

### 8.355.2 Field Documentation

8.355.2.1 `uint16_t nas_geranInstInfo::geranArfcn`

8.355.2.2 `uint8_t nas_geranInstInfo::geranBsicBcc`

8.355.2.3 `uint8_t nas_geranInstInfo::geranBsicNcc`

8.355.2.4 `int16_t nas_geranInstInfo::geranRssi`

## 8.356 nas\_gsmCellInfo Struct Reference

#### Data Fields

- `uint16_t arfcn`
- `uint8_t band1900`
- `uint8_t cellIdValid`
- `uint8_t bsicId`
- `int16_t rssi`
- `int16_t srxlev`

### 8.356.1 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 arfcn 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>

## 8.356.2 Field Documentation

8.356.2.1 uint16\_t nas\_gsmCellInfo::arfcn

8.356.2.2 uint8\_t nas\_gsmCellInfo::band1900

8.356.2.3 uint8\_t nas\_gsmCellInfo::bsicId

8.356.2.4 uint8\_t nas\_gsmCellInfo::cellIdValid

8.356.2.5 int16\_t nas\_gsmCellInfo::rsSI

8.356.2.6 int16\_t nas\_gsmCellInfo::srxlev

## 8.357 nas\_GSMRSSIThresh Struct Reference

### Data Fields

- uint8\_t [GSMRSSIThreshListLen](#)
- int16\_t \* [pGSMRSSIThreshList](#)

### 8.357.1 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>

## 8.357.2 Field Documentation

8.357.2.1 uint8\_t nas\_GSMRSSIThresh::GSMRSSIThreshListLen

8.357.2.2 int16\_t\* nas\_GSMRSSIThresh::pGSMRSSIThreshList

## 8.358 nas\_GSMsRvStatusInfo Struct Reference

## Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [trueSrvStatus](#)
- uint8\_t [isPrefDataPath](#)

## 8.358.1 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>
-----------------------	---

## 8.358.2 Field Documentation

8.358.2.1 `uint8_t nas_GSMSrvStatusInfo::isPrefDataPath`

8.358.2.2 `uint8_t nas_GSMSrvStatusInfo::srvStatus`

8.358.2.3 `uint8_t nas_GSMSrvStatusInfo::trueSrvStatus`

## 8.359 nas\_GSMSysInfo Struct Reference

### Data Fields

- [nas\\_sysInfoCommon sysInfoGSM](#)
- `uint8_t lacValid`
- `uint16_t lac`
- `uint8_t cellIdValid`
- `uint32_t cellId`
- `uint8_t regRejectInfoValid`
- `uint8_t rejectSrvDomain`
- `uint8_t rejCause`
- `uint8_t networkIdValid`
- `uint8_t MCC` [3]
- `uint8_t MNC` [3]
- `uint8_t egprsSuppValid`
- `uint8_t egprsSupp`
- `uint8_t dtmSuppValid`
- `uint8_t dtmSupp`

### 8.359.1 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>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> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 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>

## 8.359.2 Field Documentation

8.359.2.1 `uint32_t nas_GSMsSysInfo::cellId`

8.359.2.2 `uint8_t nas_GSMsSysInfo::cellIdValid`

8.359.2.3 `uint8_t nas_GSMsSysInfo::dtmSupp`

8.359.2.4 `uint8_t nas_GSMsSysInfo::dtmSuppValid`

8.359.2.5 `uint8_t nas_GSMsSysInfo::egprsSupp`

8.359.2.6 `uint8_t nas_GSMsSysInfo::egprsSuppValid`

8.359.2.7 `uint16_t nas_GSMsSysInfo::lac`

8.359.2.8 `uint8_t nas_GSMsSysInfo::lacValid`

8.359.2.9 `uint8_t nas_GSMsSysInfo::MCC[3]`

8.359.2.10 `uint8_t nas_GSMsSysInfo::MNC[3]`

8.359.2.11 `uint8_t nas_GSMsSysInfo::networkIdValid`

8.359.2.12 `uint8_t nas_GSMsSysInfo::regRejectInfoValid`

8.359.2.13 `uint8_t nas_GSMsysInfo::rejCause`

8.359.2.14 `uint8_t nas_GSMsysInfo::rejectSrvDomain`

8.359.2.15 `nas_sysInfoCommon nas_GSMsysInfo::sysInfoGSM`

## 8.360 nas\_HDRECIOTthresh Struct Reference

### Data Fields

- `uint8_t HDRECIOTthreshListLen`
- `int16_t * pHRECIOTthreshList`

### 8.360.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

#### Parameters

<i>HDRECIOTthreshListLen</i>	<ul style="list-style-type: none"><li>• Length of the HDR ECIO threshold list parameter to follow</li></ul>
<i>pHDRECIOTthreshList</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>

### 8.360.2 Field Documentation

8.360.2.1 `uint8_t nas_HDRECIOTthresh::HDRECIOTthreshListLen`

8.360.2.2 `int16_t* nas_HDRECIOTthresh::pHDRECIOTthreshList`

## 8.361 nas\_HDRIOTthresh Struct Reference

### Data Fields

- `uint8_t HDRIOTthreshListLen`
- `int16_t * pHRIOTthreshList`

### 8.361.1 Detailed Description

This structure contains HDR IO threshold related parameters.

## Parameters

<i>HDRIOTresh- ListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR IO threshold list parameter to follow</li> </ul>
<i>pHDRIOTresh- 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>

## 8.361.2 Field Documentation

8.361.2.1 `uint8_t nas_HDRIOTresh::HDRIOTreshListLen`8.361.2.2 `int16_t* nas_HDRIOTresh::pHDRIOTreshList`

## 8.362 nas\_HDRRSSIThresh Struct Reference

## Data Fields

- `uint8_t HDRRSSIThreshListLen`
- `int16_t * pHDRRSSIThreshList`

## 8.362.1 Detailed Description

This structure contains HDR RSSI threshold related parameters.

## Parameters

<i>HDRRSSI- ThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR RSSI threshold list parameter to follow</li> </ul>
<i>pHDRRSSI- 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: -118 to -13 (in dBm).</li> </ul>

## 8.362.2 Field Documentation

8.362.2.1 `uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen`8.362.2.2 `int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList`

## 8.363 nas\_HDRSINRThreshold Struct Reference

## Data Fields

- `uint8_t HDRSINRThreshListLen`
- `uint16_t * pHDRSINRThreshList`

### 8.363.1 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>

### 8.363.2 Field Documentation

8.363.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.363.2.2 `uint16_t* nas_HDRSINRThreshold::pHDRSINRThreshList`

## 8.364 nas\_HDRSysInfo Struct Reference

### Data Fields

- [nas\\_sysInfoCommon sysInfoHDR](#)
- `uint8_t isSysPrIMatchValid`
- `uint8_t isSysPrIMatch`
- `uint8_t hdrPersonalityValid`
- `uint8_t hdrPersonality`
- `uint8_t hdrActiveProtValid`
- `uint8_t hdrActiveProt`
- `uint8_t is856SysIdValid`
- `uint8_t is856SysId [16]`

### 8.364.1 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</i> [SLQ-S_SYSTEM_ID_SIZE]	<ul style="list-style-type: none"> <li>• IS-856 system ID.</li> <li>• Only applicable for HDR.</li> </ul>
--	---

### 8.364.2 Field Documentation

- 8.364.2.1 uint8\_t nas\_HDRSysInfo::hdrActiveProt
- 8.364.2.2 uint8\_t nas\_HDRSysInfo::hdrActiveProtValid
- 8.364.2.3 uint8\_t nas\_HDRSysInfo::hdrPersonality
- 8.364.2.4 uint8\_t nas\_HDRSysInfo::hdrPersonalityValid
- 8.364.2.5 uint8\_t nas\_HDRSysInfo::is856SysId[16]
- 8.364.2.6 uint8\_t nas\_HDRSysInfo::is856SysIdValid
- 8.364.2.7 uint8\_t nas\_HDRSysInfo::isSysPrIMatch
- 8.364.2.8 uint8\_t nas\_HDRSysInfo::isSysPrIMatchValid
- 8.364.2.9 nas\_sysInfoCommon nas\_HDRSysInfo::sysInfoHDR

## 8.365 nas\_infoInterFreq Struct Reference

### Data Fields

- uint16\_t [earfcn](#)
- uint8\_t [threshXLow](#)
- uint8\_t [threshXHigh](#)
- uint8\_t [cell\\_resel\\_priority](#)
- uint8\_t [cells\\_len](#)
- [nas\\_cellParams](#) [cellInterFreqParams](#) [255]

### 8.365.1 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 <code>thresh_serving_low</code>, 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[NAS_MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_cellParams</a> for more information.</li> </ul>

### 8.365.2 Field Documentation

8.365.2.1 `uint8_t nas_infoInterFreq::cell_resel_priority`

8.365.2.2 `nas_cellParams nas_infoInterFreq::cellInterFreqParams[255]`

8.365.2.3 `uint8_t nas_infoInterFreq::cells_len`

8.365.2.4 `uint16_t nas_infoInterFreq::earfcn`

8.365.2.5 `uint8_t nas_infoInterFreq::threshXHigh`

8.365.2.6 `uint8_t nas_infoInterFreq::threshXLow`

## 8.366 nas\_lteGsmCellInfo Struct Reference

### Data Fields

- `uint8_t cellReselPriority`
- `uint8_t threshGsmHigh`
- `uint8_t threshGsmLow`
- `uint8_t nccPermitted`
- `uint8_t cells_len`
- `nas_gsmCellInfo GsmCellInfo [255]`

### 8.366.1 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 ue_in_idle 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 ue_in_idle 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 ue_in_idle 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 n set to 1 means a neighbor with NCC n must be included in the report. This flag is synonymous with a blacklist in other RATs.</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 gsm cells.</li> </ul>
<i>GsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_gsmCellInfo</a> for more information.</li> </ul>

## 8.366.2 Field Documentation

8.366.2.1 uint8\_t nas\_lteGsmCellInfo::cellReselPriority

8.366.2.2 uint8\_t nas\_lteGsmCellInfo::cells\_len

8.366.2.3 nas\_gsmCellInfo nas\_lteGsmCellInfo::GsmCellInfo[255]

8.366.2.4 uint8\_t nas\_lteGsmCellInfo::nccPermitted

8.366.2.5 uint8\_t nas\_lteGsmCellInfo::threshGsmHigh

8.366.2.6 uint8\_t nas\_lteGsmCellInfo::threshGsmLow

## 8.367 nas\_LTEInfo Struct Reference

## Data Fields

- uint8\_t [band](#)
- uint8\_t [bandwidth](#)
- uint16\_t [RXChan](#)
- uint16\_t [TXChan](#)

- uint8\_t [emmState](#)
- uint8\_t [emmSubState](#)
- uint8\_t [emmConnState](#)

### 8.367.1 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> <li>– 0x07 - IMSI Detach</li> <li>– 0x08 - Waiting for ESM</li> </ul> </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>

## 8.367.2 Field Documentation

8.367.2.1 `uint8_t nas_LTEInfo::band`

8.367.2.2 `uint8_t nas_LTEInfo::bandwidth`

8.367.2.3 `uint8_t nas_LTEInfo::emmConnState`

8.367.2.4 `uint8_t nas_LTEInfo::emmState`

8.367.2.5 `uint8_t nas_LTEInfo::emmSubState`

8.367.2.6 `uint16_t nas_LTEInfo::RXChan`

8.367.2.7 `uint16_t nas_LTEInfo::TXChan`

## 8.368 nas\_LTEInfoInterfreq Struct Reference

### Data Fields

- uint8\_t [ueInIdle](#)
- uint8\_t [freqsLen](#)
- [nas\\_infoInterFreq](#) [InfoInterfreq](#) [255]

### 8.368.1 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[NA-S_MAX_DESC-RIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_infoInterFreq</a> for more information.</li> </ul>

### 8.368.2 Field Documentation

8.368.2.1 uint8\_t nas\_LTEInfoInterfreq::freqsLen

8.368.2.2 nas\_infoInterFreq nas\_LTEInfoInterfreq::InfoInterfreq[255]

8.368.2.3 uint8\_t nas\_LTEInfoInterfreq::ueInIdle

## 8.369 nas\_LTEInfoIntrafreq Struct Reference

### Data Fields

- uint8\_t [ueInIdle](#)
- uint8\_t [plmn](#) [3]
- uint16\_t [tac](#)
- uint32\_t [globalCellId](#)
- uint16\_t [earfcn](#)
- uint16\_t [servingCellId](#)
- uint8\_t [cellReselPriority](#)
- uint8\_t [sNonIntraSearch](#)
- uint8\_t [threshServingLow](#)
- uint8\_t [sIntraSearch](#)
- uint8\_t [cellsLen](#)
- [nas\\_cellParams](#) [CellParams](#) [255]

## 8.369.1 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. <ul style="list-style-type: none"> <li>– 0xFF - Not Available</li> </ul> </li> </ul>
<i>plmn[NAS_PLM-N_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 <i>ue_in_idle</i> 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 <i>ue_in_idle</i> 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 <i>ue_in_idle</i> 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_cellParams</a> for more information.</li> </ul>

## 8.369.2 Field Documentation

8.369.2.1 `nas_cellParams` `nas_LTEInfoIntrafreq::CellParams[255]`

8.369.2.2 `uint8_t` `nas_LTEInfoIntrafreq::cellReselPriority`

8.369.2.3 `uint8_t` `nas_LTEInfoIntrafreq::cellsLen`

8.369.2.4 `uint16_t` `nas_LTEInfoIntrafreq::earfcn`

8.369.2.5 `uint32_t` `nas_LTEInfoIntrafreq::globalCellId`

8.369.2.6 `uint8_t` `nas_LTEInfoIntrafreq::plmn[3]`

8.369.2.7 `uint16_t` `nas_LTEInfoIntrafreq::servingCellId`

8.369.2.8 `uint8_t` `nas_LTEInfoIntrafreq::sIntraSearch`

8.369.2.9 `uint8_t` `nas_LTEInfoIntrafreq::sNonIntraSearch`

8.369.2.10 `uint16_t` `nas_LTEInfoIntrafreq::tac`

8.369.2.11 `uint8_t` `nas_LTEInfoIntrafreq::threshServingLow`

8.369.2.12 `uint8_t` `nas_LTEInfoIntrafreq::ueInIdle`

## 8.370 nas\_LTEInfoNeighboringGSM Struct Reference

### Data Fields

- `uint8_t` [ueInIdle](#)
- `uint8_t` [freqsLen](#)
- [nas\\_lteGsmCellInfo](#) [LteGsmCellInfo](#) [255]

### 8.370.1 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. <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 GSM cell information.</li> <li>• If 0(zero), then no information follows it.</li> </ul>
<i>LteGsmCellInfo[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteGsmCellInfo</a> for more information.</li> </ul>

### 8.370.2 Field Documentation

8.370.2.1 `uint8_t nas_LTEInfoNeighboringGSM::freqsLen`

8.370.2.2 `nas_LteGsmCellInfo nas_LTEInfoNeighboringGSM::LteGsmCellInfo[255]`

8.370.2.3 `uint8_t nas_LTEInfoNeighboringGSM::ueIdle`

## 8.371 nas\_LTEInfoNeighboringWCDMA Struct Reference

### Data Fields

- `uint8_t ueIdle`
- `uint8_t freqsLen`
- `nas_LteWcdmaCellInfo LTEWCDMACellInfo [255]`

### 8.371.1 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteWcdmaCellInfo</a> for more information.</li> </ul>

### 8.371.2 Field Documentation

8.371.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.371.2.2 `nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.371.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::uelIdle`

## 8.372 nas\_lteRsrpinformation Struct Reference

### Data Fields

- `int16_t rsrplevel`

### 8.372.1 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>
------------------	--

### 8.372.2 Field Documentation

8.372.2.1 `int16_t nas_lteRsrpinformation::rsrplevel`

## 8.373 nas\_LTERSRPThresh Struct Reference

### Data Fields

- `uint8_t LTERSRPThreshListLen`
- `int16_t * pLTERSRPThreshList`

### 8.373.1 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>

### 8.373.2 Field Documentation

8.373.2.1 `uint8_t nas_LTERSRPThresh::LTERSRPThreshListLen`

8.373.2.2 `int16_t* nas_LTERSRPThresh::pLTERSRPThreshList`

## 8.374 nas\_LTERSRQThresh Struct Reference

### Data Fields

- `uint8_t LTERSRQThreshListLen`
- `int16_t* pLTERSRQThreshList`

#### 8.374.1 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>

#### 8.374.2 Field Documentation

8.374.2.1 `uint8_t nas_LTERSRQThresh::LTERSRQThreshListLen`

8.374.2.2 `int16_t* nas_LTERSRQThresh::pLTERSRQThreshList`

## 8.375 nas\_LTERSSIThresh Struct Reference

### Data Fields

- `uint8_t LTERSSIThreshListLen`
- `int16_t* pLTERSSIThreshList`

#### 8.375.1 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>

## 8.375.2 Field Documentation

8.375.2.1 uint8\_t nas\_LTERSSIThresh::LTERSSIThreshListLen

8.375.2.2 int16\_t\* nas\_LTERSSIThresh::pLTERSSIThreshList

## 8.376 nas\_LTESigRptConfig Struct Reference

### Data Fields

- uint8\_t [rptRate](#)
- uint8\_t [avgPeriod](#)

### 8.376.1 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>

## 8.376.2 Field Documentation

8.376.2.1 uint8\_t nas\_LTESigRptConfig::avgPeriod

8.376.2.2 uint8\_t nas\_LTESigRptConfig::rptRate

## 8.377 nas\_IteSnrinformation Struct Reference

## Data Fields

- `int16_t` [snrlevel](#)

## 8.377.1 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>
-----------------	---

## 8.377.2 Field Documentation

8.377.2.1 `int16_t` `nas_lteSnrinformation::snrlevel`

## 8.378 nas\_LTESNRThreshold Struct Reference

## Data Fields

- `uint8_t` [LTESNRThreshListLen](#)
- `int16_t *` [pLTESNRThreshList](#)

## 8.378.1 Detailed Description

This structure contains LTE SNR threshold related parameters.

## Parameters

<i>LTESNRThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the LTE SNR threshold list parameter to follow</li> </ul>
<i>pLTESNRThreshList</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>

## 8.378.2 Field Documentation

8.378.2.1 `uint8_t` `nas_LTESNRThreshold::LTESNRThreshListLen`

8.378.2.2 `int16_t*` `nas_LTESNRThreshold::pLTESNRThreshList`

## 8.379 nas\_LTESysInfo Struct Reference

## Data Fields

- [nas\\_sysInfoCommon sysInfoLTE](#)

- uint8\_t [lacValid](#)
- uint16\_t [lac](#)
- uint8\_t [cellIdValid](#)
- uint32\_t [cellId](#)
- uint8\_t [regRejectInfoValid](#)
- uint8\_t [rejectSrvDomain](#)
- uint8\_t [rejCause](#)
- uint8\_t [networkIdValid](#)
- uint8\_t [MCC](#) [3]
- uint8\_t [MNC](#) [3]
- uint8\_t [tacValid](#)
- uint16\_t [tac](#)

### 8.379.1 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>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>
<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> <li>In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 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>

## 8.379.2 Field Documentation

8.379.2.1 `uint32_t nas_LTESysInfo::cellId`

8.379.2.2 `uint8_t nas_LTESysInfo::cellIdValid`

8.379.2.3 `uint16_t nas_LTESysInfo::lac`

8.379.2.4 `uint8_t nas_LTESysInfo::lacValid`

- 8.379.2.5 `uint8_t nas_LTESysInfo::MCC[3]`
- 8.379.2.6 `uint8_t nas_LTESysInfo::MNC[3]`
- 8.379.2.7 `uint8_t nas_LTESysInfo::networkIdValid`
- 8.379.2.8 `uint8_t nas_LTESysInfo::regRejectInfoValid`
- 8.379.2.9 `uint8_t nas_LTESysInfo::rejCause`
- 8.379.2.10 `uint8_t nas_LTESysInfo::rejectSrvDomain`
- 8.379.2.11 `nas_sysInfoCommon nas_LTESysInfo::sysInfoLTE`
- 8.379.2.12 `uint16_t nas_LTESysInfo::tac`
- 8.379.2.13 `uint8_t nas_LTESysInfo::tacValid`

## 8.380 `nas_lteWcdmaCellInfo` Struct Reference

### Data Fields

- `uint16_t uarfcn`
- `uint8_t cellReselPriority`
- `uint16_t threshXhigh`
- `uint16_t threshXlow`
- `uint8_t cellsLen`
- `nas_wcdmaCellInfo WCDMACellInfo` [255]

### 8.380.1 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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 <code>ue_in_idle</code> 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>See <a href="#">wcdmaCellInfo</a> for more information.</li> </ul>

## 8.380.2 Field Documentation

8.380.2.1 `uint8_t nas_lteWcdmaCellInfo::cellReselPriority`

8.380.2.2 `uint8_t nas_lteWcdmaCellInfo::cellsLen`

8.380.2.3 `uint16_t nas_lteWcdmaCellInfo::threshXhigh`

8.380.2.4 `uint16_t nas_lteWcdmaCellInfo::threshXlow`

8.380.2.5 `uint16_t nas_lteWcdmaCellInfo::uarfcn`

8.380.2.6 `nas_wcdmaCellInfo nas_lteWcdmaCellInfo::WCDMACellInfo[255]`

## 8.381 nas\_MNRInfo Struct Reference

### Data Fields

- `uint16_t` [mcc](#)
- `uint16_t` [mnc](#)
- `uint32_t` [rat](#)

### 8.381.1 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>

### 8.381.2 Field Documentation

8.381.2.1 uint16\_t nas\_MNRInfo::mcc

8.381.2.2 uint16\_t nas\_MNRInfo::mnc

8.381.2.3 uint32\_t nas\_MNRInfo::rat

## 8.382 nas\_netSelectionPref Struct Reference

### Data Fields

- uint8\_t [netReg](#)
- uint16\_t [mcc](#)
- uint16\_t [mnc](#)

### 8.382.1 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>

### 8.382.2 Field Documentation

8.382.2.1 uint16\_t nas\_netSelectionPref::mcc

8.382.2.2 uint16\_t nas\_netSelectionPref::mnc

8.382.2.3 uint8\_t nas\_netSelectionPref::netReg

## 8.383 nas\_nmrCellInfo Struct Reference

### Data Fields

- uint32\_t [nmrCellID](#)
- uint8\_t [nmrPlmn](#) [3]
- uint16\_t [nmrLac](#)
- uint16\_t [nmrArfcn](#)
- uint8\_t [nmrBsic](#)
- uint16\_t [nmrRxLev](#)

### 8.383.1 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[NAS_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>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>

### 8.383.2 Field Documentation

8.383.2.1 uint16\_t nas\_nmrCellInfo::nmrArfcn

8.383.2.2 uint8\_t nas\_nmrCellInfo::nmrBsic

8.383.2.3 uint32\_t nas\_nmrCellInfo::nmrCellID

8.383.2.4 uint16\_t nas\_nmrCellInfo::nmrLac

8.383.2.5 uint8\_t nas\_nmrCellInfo::nmrPlmn[3]

8.383.2.6 uint16\_t nas\_nmrCellInfo::nmrRxLev

## 8.384 nas\_PhyCaAggPcellInfo Struct Reference

### Data Fields

- uint16\_t [pci](#)
- uint16\_t [freq](#)
- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) [dl\\_bw\\_value](#)
- uint16\_t [iLTEbandValue](#)
- uint8\_t [TlvPresent](#)

### 8.384.1 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_LITE</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_LITE</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.384.2 Field Documentation

8.384.2.1 [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) [nas\\_PhyCaAggPcellInfo::dl\\_bw\\_value](#)

8.384.2.2 [uint16\\_t](#) [nas\\_PhyCaAggPcellInfo::freq](#)

8.384.2.3 [uint16\\_t](#) [nas\\_PhyCaAggPcellInfo::iLTEbandValue](#)

8.384.2.4 [uint16\\_t](#) [nas\\_PhyCaAggPcellInfo::pci](#)

8.384.2.5 [uint8\\_t](#) [nas\\_PhyCaAggPcellInfo::TlvPresent](#)

## 8.385 nas\_PhyCaAggScellIDIBw Struct Reference

### Data Fields

- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) [dl\\_bw\\_value](#)

- uint8\_t [TlvPresent](#)

### 8.385.1 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_LITE</a> for more information.</li></ul>
--------------------	--

### 8.385.2 Field Documentation

8.385.2.1 [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) nas\_PhyCaAggScellIDBw::dl\_bw\_value

8.385.2.2 uint8\_t nas\_PhyCaAggScellIDBw::TlvPresent

## 8.386 nas\_PhyCaAggScellIndex Struct Reference

### Data Fields

- uint8\_t [scell\\_idx](#)
- uint8\_t [TlvPresent](#)

### 8.386.1 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>

### 8.386.2 Field Documentation

8.386.2.1 uint8\_t nas\_PhyCaAggScellIndex::scell\_idx

8.386.2.2 uint8\_t nas\_PhyCaAggScellIndex::TlvPresent

## 8.387 nas\_PhyCaAggScellIndType Struct Reference

### Data Fields

- uint16\_t [pci](#)
- uint16\_t [freq](#)

- [NAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_LITE](#) *scell\_state*
- `uint8_t` *TlvPresent*

### 8.387.1 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_LITE</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.387.2 Field Documentation

8.387.2.1 `uint16_t` *nas\_PhyCaAggScellIndType::freq*

8.387.2.2 `uint16_t` *nas\_PhyCaAggScellIndType::pci*

8.387.2.3 `NAS_LTE_CPHY_SCELL_STATE_LITE` *nas\_PhyCaAggScellIndType::scell\_state*

8.387.2.4 `uint8_t` *nas\_PhyCaAggScellIndType::TlvPresent*

## 8.388 nas\_PhyCaAggScellInfo Struct Reference

#### Data Fields

- `uint16_t` *pci*
- `uint16_t` *freq*
- `NAS_LTE_CPHY_CA_BW_NRB_LITE` *dl\_bw\_value*
- `uint16_t` *iLTEbandValue*
- `NAS_LTE_CPHY_SCELL_STATE_LITE` *scell\_state*
- `uint8_t` *TlvPresent*

### 8.388.1 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_LITE</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. <ul style="list-style-type: none"> <li>– 120 - LTE E-UTRA Operating Band 1</li> <li>– 121 - LTE E-UTRA Operating Band 2</li> <li>– 122 - LTE E-UTRA Operating Band 3</li> <li>– 123 - LTE E-UTRA Operating Band 4</li> <li>– 124 - LTE E-UTRA Operating Band 5</li> <li>– 125 - LTE E-UTRA Operating Band 6</li> <li>– 126 - LTE E-UTRA Operating Band 7</li> <li>– 127 - LTE E-UTRA Operating Band 8</li> <li>– 128 - LTE E-UTRA Operating Band 9</li> <li>– 129 - LTE E-UTRA Operating Band 10</li> <li>– 130 - LTE E-UTRA Operating Band 11</li> <li>– 131 - LTE E-UTRA Operating Band 12</li> <li>– 132 - LTE E-UTRA Operating Band 13</li> <li>– 133 - LTE E-UTRA Operating Band 14</li> <li>– 134 - LTE E-UTRA Operating Band 17</li> <li>– 135 - LTE E-UTRA Operating Band 33</li> <li>– 136 - LTE E-UTRA Operating Band 34</li> <li>– 137 - LTE E-UTRA Operating Band 35</li> <li>– 138 - LTE E-UTRA Operating Band 36</li> <li>– 139 - LTE E-UTRA Operating Band 37</li> <li>– 140 - LTE E-UTRA Operating Band 38</li> <li>– 141 - LTE E-UTRA Operating Band 39</li> <li>– 142 - LTE E-UTRA Operating Band 40</li> <li>– 143 - LTE E-UTRA Operating Band 18</li> <li>– 144 - LTE E-UTRA Operating Band 19</li> <li>– 145 - LTE E-UTRA Operating Band 20</li> <li>– 146 - LTE E-UTRA Operating Band 21</li> <li>– 147 - LTE E-UTRA Operating Band 24</li> <li>– 148 - LTE E-UTRA Operating Band 25</li> <li>– 149 - LTE E-UTRA Operating Band 41</li> <li>– 150 - LTE E-UTRA Operating Band 42</li> <li>– 151 - LTE E-UTRA Operating Band 43</li> <li>– 152 - LTE E-UTRA Operating Band 23</li> <li>– 153 - LTE E-UTRA Operating Band 26</li> <li>– 154 - LTE E-UTRA Operating Band 32</li> <li>– 155 - LTE E-UTRA Operating Band 125</li> <li>– 156 - LTE E-UTRA Operating Band 126</li> <li>– 157 - LTE E-UTRA Operating Band 127</li> <li>– 158 - LTE E-UTRA Operating Band 28</li> <li>– 159 - LTE E-UTRA Operating Band 29</li> <li>– 160 - LTE E-UTRA Operating Band 30</li> </ul> </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_LITE</a> for more information.</li> </ul>

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
-------------------	--

## 8.388.2 Field Documentation

8.388.2.1 **NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE** nas\_PhyCaAggScellInfo::dl\_bw\_value

8.388.2.2 uint16\_t nas\_PhyCaAggScellInfo::freq

8.388.2.3 uint16\_t nas\_PhyCaAggScellInfo::ltebandValue

8.388.2.4 uint16\_t nas\_PhyCaAggScellInfo::pci

8.388.2.5 **NAS\_LTE\_CPHY\_CELL\_STATE\_LITE** nas\_PhyCaAggScellInfo::cell\_state

8.388.2.6 uint8\_t nas\_PhyCaAggScellInfo::TlvPresent

## 8.389 nas\_qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- uint8\_t [leapSeconds](#)
- uint8\_t [localTimeOffset](#)
- uint8\_t [daylightSavings](#)

### 8.389.1 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>

## 8.389.2 Field Documentation

8.389.2.1 uint8\_t nas\_qaQmi3Gpp2TimeZone::daylightSavings

8.389.2.2 uint8\_t nas\_qaQmi3Gpp2TimeZone::leapSeconds

8.389.2.3 uint8\_t nas\_qaQmi3Gpp2TimeZone::localTimeOffset

## 8.390 nas\_QmiNas3GppNetworkInfo Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint32\_t [InUse](#)
- uint32\_t [Roaming](#)
- uint32\_t [Forbidden](#)
- uint32\_t [Preferred](#)
- char [Description](#) [255]

### 8.390.1 Detailed Description

#### Parameters

<i>MCC</i>	Mobile Country Code
<i>MNC</i>	Mobile Network Code
<i>InUse</i>	current network or not
<i>Roaming</i>	Home/Roam Status of the Network
<i>Forbidden</i>	Network in the forbidden network list or not
<i>Preferred</i>	Network in the Preferred network list or not
<i>Description</i>	Network Name/Description

### 8.390.2 Field Documentation

8.390.2.1 char nas\_QmiNas3GppNetworkInfo::Description[255]

8.390.2.2 uint32\_t nas\_QmiNas3GppNetworkInfo::Forbidden

8.390.2.3 uint32\_t nas\_QmiNas3GppNetworkInfo::InUse

8.390.2.4 uint16\_t nas\_QmiNas3GppNetworkInfo::MCC

8.390.2.5 uint16\_t nas\_QmiNas3GppNetworkInfo::MNC

8.390.2.6 uint32\_t nas\_QmiNas3GppNetworkInfo::Preferred

8.390.2.7 uint32\_t nas\_QmiNas3GppNetworkInfo::Roaming

## 8.391 nas\_QmiNas3GppNetworkRAT Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [RAT](#)

### 8.391.1 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> <li>– 0x08 - LTE</li> <li>– 0x09 - TD-SCDMA</li> </ul> </li> </ul>

## 8.391.2 Field Documentation

8.391.2.1 uint16\_t nas\_QmiNas3GppNetworkRAT::MCC

8.391.2.2 uint16\_t nas\_QmiNas3GppNetworkRAT::MNC

8.391.2.3 uint8\_t nas\_QmiNas3GppNetworkRAT::RAT

## 8.392 nas\_QmisNasPcsDigit Struct Reference

## Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [includes\\_pcs\\_digit](#)

## 8.392.1 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>

## 8.392.2 Field Documentation

8.392.2.1 uint8\_t nas\_QmisNasPcsDigit::includes\_pcs\_digit

8.392.2.2 uint16\_t nas\_QmisNasPcsDigit::MCC

8.392.2.3 uint16\_t nas\_QmisNasPcsDigit::MNC

## 8.393 nas\_RejectReasonTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [serviceDomain](#)
- uint32\_t [rejectCause](#)

### 8.393.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>serviceDomain</i>	service domain
<i>rejectCause</i>	cause of reject

### 8.393.2 Field Documentation

8.393.2.1 uint32\_t nas\_RejectReasonTlv::rejectCause

8.393.2.2 uint32\_t nas\_RejectReasonTlv::serviceDomain

8.393.2.3 uint8\_t nas\_RejectReasonTlv::TlvPresent

## 8.394 nas\_RFInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [radioInterfaceSize](#)
- uint32\_t [radioInterface](#) [255]
- uint32\_t [activeBandClass](#) [255]
- uint32\_t [activeChannel](#) [255]

### 8.394.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>radioInterface</i>	radio interface technology of the signal being measured
<i>activeBandClass</i>	active band class
<i>activeChannel</i>	active channel

### 8.394.2 Field Documentation

8.394.2.1 uint32\_t nas\_RFInfoTlv::activeBandClass[255]

8.394.2.2 uint32\_t nas\_RFInfoTlv::activeChannel[255]

8.394.2.3 uint32\_t nas\_RFInfoTlv::radiolInterface[255]

8.394.2.4 uint8\_t nas\_RFInfoTlv::radiolInterfaceSize

8.394.2.5 uint8\_t nas\_RFInfoTlv::TlvPresent

## 8.395 nas\_roamIndList Struct Reference

### Data Fields

- uint8\_t [numInstances](#)
- uint8\_t [radiolInterface](#) [32]
- uint8\_t [roamIndicator](#) [32]

### 8.395.1 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>

### 8.395.2 Field Documentation

8.395.2.1 uint8\_t nas\_roamIndList::numInstances

8.395.2.2 uint8\_t nas\_roamIndList::radiolInterface[32]

8.395.2.3 `uint8_t nas_roamIndList::roamIndicator[32]`

## 8.396 `nas_rsrqInformation` Struct Reference

### Data Fields

- `int8_t rsrq`
- `uint8_t radiolf`

### 8.396.1 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>

### 8.396.2 Field Documentation

8.396.2.1 `uint8_t nas_rsrqInformation::radiolf`

8.396.2.2 `int8_t nas_rsrqInformation::rsrq`

## 8.397 `nas_rxSignalStrengthListElement` Struct Reference

### Data Fields

- `int16_t rxSignalStrength`
- `uint8_t radiolf`

### 8.397.1 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.

**8.397.2 Field Documentation**

8.397.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.397.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

**8.398 nas\_servSystem Struct Reference****Data Fields**

- `uint8_t regState`
- `uint8_t csAttachState`
- `uint8_t psAttachState`
- `uint8_t selNetwork`
- `uint8_t numRadiolInterfaces`
- `uint8_t radioInterface` [32]

**8.398.1 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 currently in use</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>

## 8.398.2 Field Documentation

8.398.2.1 `uint8_t nas_servSystem::csAttachState`

8.398.2.2 `uint8_t nas_servSystem::numRadioInterfaces`

8.398.2.3 `uint8_t nas_servSystem::psAttachState`

8.398.2.4 `uint8_t nas_servSystem::radioInterface[32]`

8.398.2.5 `uint8_t nas_servSystem::regState`

8.398.2.6 `uint8_t nas_servSystem::selNetwork`

## 8.399 nas\_SignalStrengthTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- int8\_t [signalStrength](#)
- uint32\_t [radioInterface](#)

### 8.399.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>signalStrength</i>	signal strength
<i>radioInterface</i>	radio interface technology of the signal being measured

### 8.399.2 Field Documentation

8.399.2.1 uint32\_t nas\_SignalStrengthTlv::radioInterface

8.399.2.2 int8\_t nas\_SignalStrengthTlv::signalStrength

8.399.2.3 uint8\_t nas\_SignalStrengthTlv::TlvPresent

## 8.400 nas\_SLQSSignalStrengthsIndReq Struct Reference

### Data Fields

- uint8\_t [rxSignalStrengthDelta](#)
- uint8\_t [ecioDelta](#)
- uint8\_t [ioDelta](#)
- uint8\_t [sinrDelta](#)
- uint8\_t [rsrqDelta](#)
- uint8\_t [ecioThresholdListLen](#)
- int16\_t [ecioThresholdList](#) [10]
- uint8\_t [sinrThresholdListLen](#)
- uint8\_t [sinrThresholdList](#) [5]
- uint16\_t [lteSnrDelta](#)
- uint8\_t [lteRsrpDelta](#)

### 8.400.1 Detailed Description

#### Parameters

<i>rxSignalStrengthDelta</i>	RSSI delta(in dBm) at which an event report indication
<i>ecioDelta</i>	ecio delta
<i>ioDelta</i>	io delta
<i>sinrDelta</i>	sinr delta
<i>rsrqDelta</i>	rsrq delta
<i>ecioThresholdListLen</i>	
<i>ecioThresholdList</i>	

<i>sinrThreshold-ListLen</i>	
<i>sinrThreshold-List</i>	
<i>lteSnrDelta</i>	lte snr delta
<i>lteRsrpDelta</i>	lte rsrp delta

## 8.400.2 Field Documentation

- 8.400.2.1 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioDelta`
- 8.400.2.2 `int16_t nas_SLQSSignalStrengthsIndReq::ecioThresholdList[10]`
- 8.400.2.3 `uint8_t nas_SLQSSignalStrengthsIndReq::ecioThresholdListLen`
- 8.400.2.4 `uint8_t nas_SLQSSignalStrengthsIndReq::ioDelta`
- 8.400.2.5 `uint8_t nas_SLQSSignalStrengthsIndReq::lteRsrpDelta`
- 8.400.2.6 `uint16_t nas_SLQSSignalStrengthsIndReq::lteSnrDelta`
- 8.400.2.7 `uint8_t nas_SLQSSignalStrengthsIndReq::rsrqDelta`
- 8.400.2.8 `uint8_t nas_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta`
- 8.400.2.9 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrDelta`
- 8.400.2.10 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdList[5]`
- 8.400.2.11 `uint8_t nas_SLQSSignalStrengthsIndReq::sinrThresholdListLen`

## 8.401 nas\_SLQSSignalStrengthsInformation Struct Reference

### Data Fields

- [nas\\_rxSignalStrengthListElement rxSignalStrengthInfo](#)
- [nas\\_ecioListElement ecioInfo](#)
- `uint32_t io`
- `uint8_t sinr`
- [nas\\_errorRateListElement errorRateInfo](#)
- [nas\\_rsrqInformation rsrqInfo](#)
- [nas\\_lteSnrinformation lteSnrinfo](#)
- [nas\\_lteRsrpinformation lteRsrpinfo](#)

### 8.401.1 Detailed Description

#### Parameters

<i>rxSignal- StrengthInfo</i>	signal strength info list
<i>ecioInfo</i>	ecio info list
<i>io</i>	received IO in dBm; IO is only applicable for 1xEV-DO
<i>sinr</i>	SINR level
<i>errorRateInfo</i>	error rate info

<i>rsrqInfo</i>	rsrq info
<i>lteSnrinfo</i>	lte Snr information
<i>lteRsrpinfo</i>	lte rsrp info

### 8.401.2 Field Documentation

8.401.2.1 `nas_ecioListElement` `nas_SLQSSignalStrengthsInformation::eciInfo`

8.401.2.2 `nas_errorRateListElement` `nas_SLQSSignalStrengthsInformation::errorRateInfo`

8.401.2.3 `uint32_t` `nas_SLQSSignalStrengthsInformation::io`

8.401.2.4 `nas_lteRsrpInformation` `nas_SLQSSignalStrengthsInformation::lteRsrpinfo`

8.401.2.5 `nas_lteSnrInformation` `nas_SLQSSignalStrengthsInformation::lteSnrinfo`

8.401.2.6 `nas_rsrqInformation` `nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.401.2.7 `nas_rxSignalStrengthListElement` `nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.401.2.8 `uint8_t` `nas_SLQSSignalStrengthsInformation::sinr`

## 8.402 nas\_SLQSSignalStrengthsTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `nas_SLQSSignalStrengthsInformation` [sSLQSSignalStrengthsInfo](#)

### 8.402.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>sSLQSSignalStrengthsInfo</i>	signal strength info

### 8.402.2 Field Documentation

8.402.2.1 `nas_SLQSSignalStrengthsInformation` `nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo`

8.402.2.2 `uint8_t` `nas_SLQSSignalStrengthsTlv::TlvPresent`

## 8.403 nas\_SrvStatusInfo Struct Reference

### Data Fields

- `uint8_t` [srvStatus](#)
- `uint8_t` [isPrefDataPath](#)

### 8.403.1 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>

### 8.403.2 Field Documentation

8.403.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`

8.403.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`

## 8.404 nas\_sysInfoCommon Struct Reference

### Data Fields

- `uint8_t srvDomainValid`
- `uint8_t srvDomain`
- `uint8_t srvCapabilityValid`
- `uint8_t srvCapability`
- `uint8_t roamStatusValid`
- `uint8_t roamStatus`
- `uint8_t isSysForbiddenValid`
- `uint8_t isSysForbidden`

### 8.404.1 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> <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 Numbers</li> <li>– 0x40 to 0xFF - Reserved.</li> <li>– 0xFF - Not Available</li> </ul> </li> <li>• Values from 0x02 onward are only applicable for 3GPP2</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>

## 8.404.2 Field Documentation

8.404.2.1 uint8\_t nas\_sysInfoCommon::isSysForbidden

8.404.2.2 uint8\_t nas\_sysInfoCommon::isSysForbiddenValid

8.404.2.3 uint8\_t nas\_sysInfoCommon::roamStatus

8.404.2.4 uint8\_t nas\_sysInfoCommon::roamStatusValid

8.404.2.5 uint8\_t nas\_sysInfoCommon::srvCapability

8.404.2.6 uint8\_t nas\_sysInfoCommon::srvCapabilityValid

8.404.2.7 uint8\_t nas\_sysInfoCommon::srvDomain

8.404.2.8 uint8\_t nas\_sysInfoCommon::srvDomainValid

## 8.405 nas\_TDSCDMAECIOThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMAECIOThreshListLen](#)
- float \* [pTDSCDMAECIOThreshList](#)

### 8.405.1 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>

### 8.405.2 Field Documentation

8.405.2.1 float\* nas\_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.405.2.2 uint8\_t nas\_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.406 nas\_TDSCDMARSCPThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMARSCPThreshListLen](#)
- int16\_t \* [pTDSCDMARSCPThreshList](#)

### 8.406.1 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>

### 8.406.2 Field Documentation

8.406.2.1 int16\_t\* nas\_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.406.2.2 `uint8_t nas_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen`

## 8.407 `nas_TDSCDMARSSIThresh` Struct Reference

### Data Fields

- `uint8_t` [TDSCDMARSSIThreshListLen](#)
- `float *` [pTDSCDMARSSIThreshList](#)

### 8.407.1 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>

### 8.407.2 Field Documentation

8.407.2.1 `float* nas_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList`

8.407.2.2 `uint8_t nas_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen`

## 8.408 `nas_TDSCDMASINRThresh` Struct Reference

### Data Fields

- `uint8_t` [TDSCDMASINRThreshListLen](#)
- `float *` [pTDSCDMASINRThreshList](#)

### 8.408.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

#### Parameters

<i>TDSCDMASINRThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the TDSCDMA SINR threshold list parameter to follow</li> </ul>
<i>pTDSCDMASINRThreshList</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>

### 8.408.2 Field Documentation

8.408.2.1 float\* nas\_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.408.2.2 uint8\_t nas\_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.409 nas\_timelInfo Struct Reference

### Data Fields

- uint16\_t [year](#)
- uint8\_t [month](#)
- uint8\_t [day](#)
- uint8\_t [hour](#)
- uint8\_t [minute](#)
- uint8\_t [second](#)
- uint8\_t [dayOfWeek](#)
- int8\_t [timeZone](#)
- uint8\_t [dayLtSavingAdj](#)
- uint8\_t [radioInterface](#)
- uint8\_t [TlvPresent](#)

### 8.409.1 Detailed Description

This structure contains the parameters for Network Time.

#### 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</li><li>• 1 is January and 12 is December</li></ul>
<i>day</i>	<ul style="list-style-type: none"><li>• Day</li><li>• Range - 1 to 31</li></ul>
<i>hour</i>	<ul style="list-style-type: none"><li>• Hour</li><li>• Range - 0 to 59</li></ul>
<i>minute</i>	<ul style="list-style-type: none"><li>• Minute</li><li>• Range - 0 to 59</li></ul>
<i>second</i>	<ul style="list-style-type: none"><li>• Second</li><li>• Range - 0 to 59</li></ul>
<i>dayOfWeek</i>	<ul style="list-style-type: none"><li>• Day of the week</li><li>• 0 is Monday and 6 is Sunday</li></ul>

<i>timeZone</i>	<ul style="list-style-type: none"> <li>• Offset from Universal time</li> <li>• The difference between local time and Universal time, in increments of 15 min</li> <li>• Signed Value</li> </ul>
<i>dayLtSavingAdj</i>	<ul style="list-style-type: none"> <li>• Daylight saving adjustment in hours</li> <li>• Possible values - 0, 1, and 2.</li> <li>• This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVDO</li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface from which the information comes</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1XEVDO - 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>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.409.2 Field Documentation

8.409.2.1 uint8\_t nas\_timeInfo::day

8.409.2.2 uint8\_t nas\_timeInfo::dayLtSavingAdj

8.409.2.3 uint8\_t nas\_timeInfo::dayOfWeek

8.409.2.4 uint8\_t nas\_timeInfo::hour

8.409.2.5 uint8\_t nas\_timeInfo::minute

8.409.2.6 uint8\_t nas\_timeInfo::month

8.409.2.7 uint8\_t nas\_timeInfo::radioInterface

8.409.2.8 uint8\_t nas\_timeInfo::second

8.409.2.9 int8\_t nas\_timeInfo::timeZone

8.409.2.10 uint8\_t nas\_timeInfo::TlvPresent

8.409.2.11 uint16\_t nas\_timeInfo::year

## 8.410 nas\_UMTSInfo Struct Reference

## Data Fields

- uint16\_t [cellID](#)
- uint8\_t [plmn](#) [3]
- uint16\_t [lac](#)
- uint16\_t [uarfcn](#)
- uint16\_t [psc](#)
- int16\_t [rscp](#)
- int16\_t [ecio](#)
- uint8\_t [umtsInst](#)
- [nas\\_UMTSInstInfo](#) [UMTSInstInfo](#) [255]
- uint8\_t [geranInst](#)
- [nas\\_geranInstInfo](#) [GeranInstInfo](#) [255]

## 8.410.1 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[NAS_PLM-N_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[N-AS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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[N-AS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_geranInstInfo</a> for more information.</li> </ul>

## 8.410.2 Field Documentation

8.410.2.1 `uint16_t nas_UMTSInfo::cellID`

8.410.2.2 `int16_t nas_UMTSInfo::ecio`

8.410.2.3 `uint8_t nas_UMTSInfo::geranInst`

8.410.2.4 `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`

8.410.2.5 `uint16_t nas_UMTSInfo::lac`

8.410.2.6 `uint8_t nas_UMTSInfo::plmn[3]`

8.410.2.7 `uint16_t nas_UMTSInfo::psc`

8.410.2.8 `int16_t nas_UMTSInfo::rscp`

8.410.2.9 `uint16_t nas_UMTSInfo::uarfcn`

8.410.2.10 `uint8_t nas_UMTSInfo::umtsInst`

8.410.2.11 `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

## 8.411 nas\_UMTSInstInfo Struct Reference

### Data Fields

- `uint16_t umtsUarfcn`
- `uint16_t umtsPsc`
- `int16_t umtsRscp`
- `int16_t umtsEcio`

### 8.411.1 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>

## 8.411.2 Field Documentation

8.411.2.1 int16\_t nas\_UMTSinstInfo::umtsEcio

8.411.2.2 uint16\_t nas\_UMTSinstInfo::umtsPsc

8.411.2.3 int16\_t nas\_UMTSinstInfo::umtsRscp

8.411.2.4 uint16\_t nas\_UMTSinstInfo::umtsUarfcn

## 8.412 nas\_umtsLTENbrCell Struct Reference

## Data Fields

- uint16\_t [earfcn](#)
- uint16\_t [pci](#)
- uint32\_t [rsrp](#)
- uint32\_t [rsrq](#)
- int16\_t [srxlev](#)
- uint8\_t [cellsTDD](#)

## 8.412.1 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_CEL_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>

## 8.412.2 Field Documentation

8.412.2.1 uint8\_t nas\_umtsLTENbrCell::cellsTDD

8.412.2.2 uint16\_t nas\_umtsLTENbrCell::earfcn

8.412.2.3 uint16\_t nas\_umtsLTENbrCell::pci

8.412.2.4 uint32\_t nas\_umtsLTENbrCell::rsrp

8.412.2.5 uint32\_t nas\_umtsLTENbrCell::rsrq

8.412.2.6 int16\_t nas\_umtsLTENbrCell::srxlev

## 8.413 nas\_UniversalTime Struct Reference

### Data Fields

- uint16\_t [year](#)
- uint8\_t [month](#)
- uint8\_t [day](#)
- uint8\_t [hour](#)
- uint8\_t [minute](#)
- uint8\_t [second](#)
- uint8\_t [dayOfWeek](#)

### 8.413.1 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>

### 8.413.2 Field Documentation

8.413.2.1 uint8\_t nas\_UniversalTime::day

8.413.2.2 uint8\_t nas\_UniversalTime::dayOfWeek

8.413.2.3 uint8\_t nas\_UniversalTime::hour

8.413.2.4 uint8\_t nas\_UniversalTime::minute

8.413.2.5 uint8\_t nas\_UniversalTime::month

8.413.2.6 uint8\_t nas\_UniversalTime::second

8.413.2.7 uint16\_t nas\_UniversalTime::year

## 8.414 nas\_wcdmaCellInfo Struct Reference

### Data Fields

- uint16\_t [psc](#)
- int16\_t [cpich\\_rscp](#)
- int16\_t [cpich\\_ecno](#)
- int16\_t [srxlev](#)

### 8.414.1 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>

### 8.414.2 Field Documentation

8.414.2.1 int16\_t nas\_wcdmaCellInfo::cpich\_ecno

8.414.2.2 int16\_t nas\_wcdmaCellInfo::cpich\_rscp

8.414.2.3 uint16\_t nas\_wcdmaCellInfo::psc

8.414.2.4 int16\_t nas\_wcdmaCellInfo::srxlev

## 8.415 nas\_WCDMAECIOThresh Struct Reference

### Data Fields

- uint8\_t [WCDMAECIOThreshListLen](#)
- int16\_t \* [pWCDMAECIOThreshList](#)

### 8.415.1 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>

### 8.415.2 Field Documentation

8.415.2.1 int16\_t\* nas\_WCDMAECIOThresh::pWCDMAECIOThreshList

8.415.2.2 `uint8_t nas_WCDMAECIOThresh::WCDMAECIOThreshListLen`

## 8.416 nas\_WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- `uint32_t wcdmaRRCTest`
- `uint8_t umtsLTENbrCellLen`
- `nas_umtsLTENbrCell UMTSLTENbrCell` [255]

### 8.416.1 Detailed Description

This structure contains information about the WCDMA - LTE Neighboring Cell Info Set.

#### Parameters

<i>wcdmaRRCTest</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>umtsLTENbrCellLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of UMTS LTE Neighbors.</li> </ul>
<i>UMTSLTENbrCell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_umtsLTENbrCell</a> for more information.</li> </ul>

### 8.416.2 Field Documentation

8.416.2.1 `nas_umtsLTENbrCell nas_WCDMAInfoLTENeighborCell::UMTSLTENbrCell`[255]

8.416.2.2 `uint8_t nas_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen`

8.416.2.3 `uint32_t nas_WCDMAInfoLTENeighborCell::wcdmaRRCTest`

## 8.417 nas\_WCDMARSSIThresh Struct Reference

### Data Fields

- `uint8_t WCDMARSSIThreshListLen`
- `int16_t * pWCDMARSSIThreshList`

### 8.417.1 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>

### 8.417.2 Field Documentation

8.417.2.1 `int16_t nas_WCDMARSSIThresh::pWCDMARSSIThreshList`

8.417.2.2 `uint8_t nas_WCDMARSSIThresh::WCDMARSSIThreshListLen`

## 8.418 nas\_WCDMASysInfo Struct Reference

### Data Fields

- [nas\\_sysInfoCommon sysInfoWCDMA](#)
- `uint8_t lacValid`
- `uint16_t lac`
- `uint8_t cellIdValid`
- `uint32_t cellId`
- `uint8_t regRejectInfoValid`
- `uint8_t rejectSrvDomain`
- `uint8_t rejCause`
- `uint8_t networkIdValid`
- `uint8_t MCC [3]`
- `uint8_t MNC [3]`
- `uint8_t hsCallStatusValid`
- `uint8_t hsCallStatus`
- `uint8_t hsIndValid`
- `uint8_t hsInd`
- `uint8_t pscValid`
- `uint16_t psc`

### 8.418.1 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>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>
<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_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> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.</li> </ul>
<i>hsCallStatus-Valid</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>
------------	--

## 8.418.2 Field Documentation

- 8.418.2.1 `uint32_t nas_WCDMA SysInfo::cellId`
- 8.418.2.2 `uint8_t nas_WCDMA SysInfo::cellIdValid`
- 8.418.2.3 `uint8_t nas_WCDMA SysInfo::hsCallStatus`
- 8.418.2.4 `uint8_t nas_WCDMA SysInfo::hsCallStatusValid`
- 8.418.2.5 `uint8_t nas_WCDMA SysInfo::hsInd`
- 8.418.2.6 `uint8_t nas_WCDMA SysInfo::hsIndValid`
- 8.418.2.7 `uint16_t nas_WCDMA SysInfo::lac`
- 8.418.2.8 `uint8_t nas_WCDMA SysInfo::lacValid`
- 8.418.2.9 `uint8_t nas_WCDMA SysInfo::MCC[3]`
- 8.418.2.10 `uint8_t nas_WCDMA SysInfo::MNC[3]`
- 8.418.2.11 `uint8_t nas_WCDMA SysInfo::networkIdValid`
- 8.418.2.12 `uint16_t nas_WCDMA SysInfo::psc`
- 8.418.2.13 `uint8_t nas_WCDMA SysInfo::pscValid`
- 8.418.2.14 `uint8_t nas_WCDMA SysInfo::regRejectInfoValid`
- 8.418.2.15 `uint8_t nas_WCDMA SysInfo::rejCause`
- 8.418.2.16 `uint8_t nas_WCDMA SysInfo::rejectSrvDomain`
- 8.418.2.17 `nas_sysInfoCommon nas_WCDMA SysInfo::sysInfoWCDMA`

## 8.419 NASBandPreferenceTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint64_t` [band\\_pref](#)

## 8.419.1 Field Documentation

- 8.419.1.1 `uint64_t NASBandPreferenceTlv::band_pref`

8.419.1.2 uint8\_t NASBandPreferenceTlv::TlvPresent

## 8.420 nasCellLocationInfoResp Struct Reference

### Data Fields

- [GERANInfo](#) \* [pGERANInfo](#)
- [UMTSInfo](#) \* [pUMTSInfo](#)
- [CDMAInfo](#) \* [pCDMAInfo](#)
- [LTEInfoIntrafreq](#) \* [pLTEInfoIntrafreq](#)
- [LTEInfoInterfreq](#) \* [pLTEInfoInterfreq](#)
- [LTEInfoNeighboringGSM](#) \* [pLTEInfoNeighboringGSM](#)
- [LTEInfoNeighboringWCDMA](#) \* [pLTEInfoNeighboringWCDMA](#)
- [ULONG](#) \* [pUMTSCellID](#)
- [WCDMAInfoLTENeighborCell](#) \* [pWCDMAInfoLTENeighborCell](#)

### 8.420.1 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>pWCDMAInfoLTENeighborCell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">WCDMAInfoLTENeighborCell</a> for more information.</li> </ul>

### 8.420.2 Field Documentation

- 8.420.2.1 **CDMAInfo\*** nasCellLocationInfoResp::pCDMAInfo
- 8.420.2.2 **GERANInfo\*** nasCellLocationInfoResp::pGERANInfo
- 8.420.2.3 **LTEInfoInterfreq\*** nasCellLocationInfoResp::pLTEInfoInterfreq
- 8.420.2.4 **LTEInfoIntrafreq\*** nasCellLocationInfoResp::pLTEInfoIntrafreq
- 8.420.2.5 **LTEInfoNeighboringGSM\*** nasCellLocationInfoResp::pLTEInfoNeighboringGSM
- 8.420.2.6 **LTEInfoNeighboringWCDMA\*** nasCellLocationInfoResp::pLTEInfoNeighboringWCDMA
- 8.420.2.7 **ULONG\*** nasCellLocationInfoResp::pUMTSCellID
- 8.420.2.8 **UMTSInfo\*** nasCellLocationInfoResp::pUMTSInfo
- 8.420.2.9 **WCDMAInfoLTENeighborCell\*** nasCellLocationInfoResp::pWCDMAInfoLTENeighborCell

## 8.421 NASEmergencyModeTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [EmerMode](#)

### 8.421.1 Field Documentation

- 8.421.1.1 uint8\_t NASEmergencyModeTlv::EmerMode
- 8.421.1.2 uint8\_t NASEmergencyModeTlv::TlvPresent

## 8.422 nasGet3GPP2SubscriptionInfoReq Struct Reference

### Data Fields

- [BYTE](#) *namID*

### 8.422.1 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>
--------------	---

### 8.422.2 Field Documentation

- 8.422.2.1 **BYTE** nasGet3GPP2SubscriptionInfoReq::namID

## 8.423 nasGet3GPP2SubscriptionInfoResp Struct Reference

### Data Fields

- [namName](#) \* [pNAMNameInfo](#)
- [dirNum](#) \* [pDirNum](#)
- [homeSIDNID](#) \* [pHomeSIDNID](#)
- [minBasedIMSI](#) \* [pMinBasedIMSI](#)
- [trueIMSI](#) \* [pTrueIMSI](#)
- [CDMAChannel](#) \* [pCDMAChannel](#)

### 8.423.1 Detailed Description

This structure contains the SLQSNasGet3GPP2Subscription response parameters.

#### Parameters

<i>pNAMNameInfo</i>	[Optional] • See <a href="#">namName</a> for more information
<i>pDirNum</i>	[Optional] • See <a href="#">dirNum</a> for more information
<i>pHomeSIDNID</i>	[Optional] • See <a href="#">homeSIDNID</a> for more information
<i>pMinBasedIMSI</i>	[Optional] • See <a href="#">minBasedIMSI</a> for more information
<i>pTrueIMSI</i>	[Optional] • See <a href="#">trueIMSI</a> for more information
<i>pCDMAChannel</i>	[Optional] • See <a href="#">CDMAChannel</a> for more information

### 8.423.2 Field Documentation

8.423.2.1 **CDMAChannel**\* nasGet3GPP2SubscriptionInfoResp::pCDMAChannel

8.423.2.2 **dirNum**\* nasGet3GPP2SubscriptionInfoResp::pDirNum

8.423.2.3 **homeSIDNID**\* nasGet3GPP2SubscriptionInfoResp::pHomeSIDNID

8.423.2.4 **minBasedIMSI**\* nasGet3GPP2SubscriptionInfoResp::pMinBasedIMSI

8.423.2.5 **namName**\* nasGet3GPP2SubscriptionInfoResp::pNAMNameInfo

8.423.2.6 **trueIMSI**\* nasGet3GPP2SubscriptionInfoResp::pTrueIMSI

## 8.424 nasGetHDRColorCodeResp Struct Reference

## Data Fields

- [BYTE](#) \* [pColorCode](#)

### 8.424.1 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>
-------------------	---

### 8.424.2 Field Documentation

8.424.2.1 [BYTE](#)\* [nasGetHDRColorCodeResp::pColorCode](#)

## 8.425 nasGetLTECphyCa Struct Reference

## Data Fields

- [PhyCaAggScellIndType](#) [sPhyCaAggScellIndType](#)
- [PhyCaAggScellIDBw](#) [sPhyCaAggScellIDBw](#)
- [PhyCaAggScellInfo](#) [sPhyCaAggScellInfo](#)
- [PhyCaAggPcellInfo](#) [sPhyCaAggPcellInfo](#)
- [PhyCaAggScellIndex](#) [sPhyCaAggScellIndex](#)

### 8.425.1 Field Documentation

8.425.1.1 [PhyCaAggPcellInfo](#) [nasGetLTECphyCa::sPhyCaAggPcellInfo](#)

8.425.1.2 [PhyCaAggScellIDBw](#) [nasGetLTECphyCa::sPhyCaAggScellIDBw](#)

8.425.1.3 [PhyCaAggScellIndex](#) [nasGetLTECphyCa::sPhyCaAggScellIndex](#)

8.425.1.4 [PhyCaAggScellIndType](#) [nasGetLTECphyCa::sPhyCaAggScellIndType](#)

8.425.1.5 [PhyCaAggScellInfo](#) [nasGetLTECphyCa::sPhyCaAggScellInfo](#)

## 8.426 NasGetLTECphyCaInfo Struct Reference

## Data Fields

- [NASPhyCaAggScellIndType](#) [PhyCaAggScellIndType](#)
- [NASPhyCaAggScellIDBw](#) [PhyCaAggScellIDBw](#)
- [NASPhyCaAggScellInfo](#) [PhyCaAggScellInfo](#)
- [NASPhyCaAggPcellInfo](#) [PhyCaAggPcellInfo](#)
- [NASPhyCaAggScellIndex](#) [PhyCaAggScellIndex](#)

### 8.426.1 Field Documentation

- 8.426.1.1 **NASPhyCaAggPcellInfo** NasGetLTECphyCalInfo::PhyCaAggPcellInfo
- 8.426.1.2 **NASPhyCaAggScellIDIBw** NasGetLTECphyCalInfo::PhyCaAggScellIDIBw
- 8.426.1.3 **NASPhyCaAggScellIndex** NasGetLTECphyCalInfo::PhyCaAggScellIndex
- 8.426.1.4 **NASPhyCaAggScellIndType** NasGetLTECphyCalInfo::PhyCaAggScellIndType
- 8.426.1.5 **NASPhyCaAggScellInfo** NasGetLTECphyCalInfo::PhyCaAggScellInfo

## 8.427 nasGetLTECphyCaResp Struct Reference

### Data Fields

- [PhyCaAggScellIndType](#) \* [pPhyCaAggScellIndType](#)
- [PhyCaAggScellIDIBw](#) \* [pPhyCaAggScellIDIBw](#)
- [PhyCaAggScellInfo](#) \* [pPhyCaAggScellInfo](#)
- [PhyCaAggPcellInfo](#) \* [pPhyCaAggPcellInfo](#)
- [PhyCaAggScellIndex](#) \* [pPhyCaAggScellIndex](#)

### 8.427.1 Field Documentation

- 8.427.1.1 **PhyCaAggPcellInfo\*** nasGetLTECphyCaResp::pPhyCaAggPcellInfo
- 8.427.1.2 **PhyCaAggScellIDIBw\*** nasGetLTECphyCaResp::pPhyCaAggScellIDIBw
- 8.427.1.3 **PhyCaAggScellIndex\*** nasGetLTECphyCaResp::pPhyCaAggScellIndex
- 8.427.1.4 **PhyCaAggScellIndType\*** nasGetLTECphyCaResp::pPhyCaAggScellIndType
- 8.427.1.5 **PhyCaAggScellInfo\*** nasGetLTECphyCaResp::pPhyCaAggScellInfo

## 8.428 nasGetSigInfoResp Struct Reference

### Data Fields

- [CDMASSInfo](#) \* [pCDMASSInfo](#)
- [HDRSSInfo](#) \* [pHDRSSInfo](#)
- [INT8](#) \* [pGSMSSInfo](#)
- [CDMASSInfo](#) \* [pWCDMASSInfo](#)
- [LTESSInfo](#) \* [pLTESSInfo](#)
- [INT8](#) \* [pTDSCDMASigInfoRscp](#)
- [TDSCDMASigInfoExt](#) \* [pTDSCDMASigInfoExt](#)

### 8.428.1 Detailed Description

This structure contains the SLQSNasGetSigInfo response parameters.

## Parameters

<i>pCDMASSInfo</i>	[Optional] • See <a href="#">CDMASSInfo</a> for more information
<i>pHDRSSInfo</i>	[Optional] • See <a href="#">HDRSSInfo</a> for more information
<i>pGSMSSInfo</i>	[Optional] • GSM signal strength is the RSSI in dBm (signed value). • A value of -125 dBm or lower is used to indicate No Signal.
<i>pWCDMASSInfo</i>	[Optional] • See <a href="#">CDMASSInfo</a> for more information
<i>pLTESSInfo</i>	[Optional] • See <a href="#">LTESSInfo</a> for more information
<i>pTDSCDMASig-InfoRscp</i>	[Optional] • RSCP of the Primary Common Control Physical Channel (PCCPCH) in dBm. • Measurement range: -120 dBm to -25 dBm.
<i>pTDSCDMASig-InfoExt</i>	[Optional] • See <a href="#">TDSCDMASigInfoExt</a> for more information.

## 8.428.2 Field Documentation

8.428.2.1 **CDMASSInfo\*** nasGetSigInfoResp::pCDMASSInfo8.428.2.2 **INT8\*** nasGetSigInfoResp::pGSMSSInfo8.428.2.3 **HDRSSInfo\*** nasGetSigInfoResp::pHDRSSInfo8.428.2.4 **LTESInfo\*** nasGetSigInfoResp::pLTESInfo8.428.2.5 **TDSCDMASigInfoExt\*** nasGetSigInfoResp::pTDSCDMASigInfoExt8.428.2.6 **INT8\*** nasGetSigInfoResp::pTDSCDMASigInfoRscp8.428.2.7 **CDMASSInfo\*** nasGetSigInfoResp::pWCDMASSInfo

## 8.429 nasGetSysInfoResp Struct Reference

## Data Fields

- [SrvStatusInfo](#) \* [pCDMASrvStatusInfo](#)
- [SrvStatusInfo](#) \* [pHDRSrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pGSMSrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pWCDMASrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pLTESrvStatusInfo](#)
- [CDMASysInfo](#) \* [pCDMASysInfo](#)
- [HDRSysInfo](#) \* [pHDRSysInfo](#)
- [GSMSysInfo](#) \* [pGSMSysInfo](#)
- [WCDMASysInfo](#) \* [pWCDMASysInfo](#)

- [LTESysInfo](#) \* [pLTESysInfo](#)
- [AddCDMASysInfo](#) \* [pAddCDMASysInfo](#)
- [WORD](#) \* [pAddHDRSysInfo](#)
- [AddSysInfo](#) \* [pAddGSMSysInfo](#)
- [AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [WORD](#) \* [pAddLTESysInfo](#)
- [CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [BYTE](#) \* [pLTEVoiceSupportSysInfo](#)
- [BYTE](#) \* [pGSMCipherDomainSysInfo](#)
- [BYTE](#) \* [pWCDMACipherDomainSysInfo](#)

### 8.429.1 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>pAddGSM Sys-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>

## 8.429.2 Field Documentation

8.429.2.1 **AddCDMASysInfo\*** nasGetSysInfoResp::pAddCDMASysInfo

8.429.2.2 **AddSysInfo\*** nasGetSysInfoResp::pAddGSM SysInfo

8.429.2.3 **WORD\*** nasGetSysInfoResp::pAddHDRSysInfo

8.429.2.4 **WORD\*** nasGetSysInfoResp::pAddLTESysInfo

8.429.2.5 **AddSysInfo\*** nasGetSysInfoResp::pAddWCDMASysInfo

8.429.2.6 **SrvStatusInfo\*** nasGetSysInfoResp::pCDMASrvStatusInfo

8.429.2.7 **CDMASysInfo\*** nasGetSysInfoResp::pCDMASysInfo

- 8.429.2.8 **CallBarringSysInfo\*** nasGetSysInfoResp::pGSMCallBarringSysInfo
- 8.429.2.9 **BYTE\*** nasGetSysInfoResp::pGSMCipherDomainSysInfo
- 8.429.2.10 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pGSMSrvStatusInfo
- 8.429.2.11 **GSMSysInfo\*** nasGetSysInfoResp::pGSMSysInfo
- 8.429.2.12 **SrvStatusInfo\*** nasGetSysInfoResp::pHRSrvStatusInfo
- 8.429.2.13 **HDRSysInfo\*** nasGetSysInfoResp::pHRSysInfo
- 8.429.2.14 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pLTESrvStatusInfo
- 8.429.2.15 **LTESysInfo\*** nasGetSysInfoResp::pLTESysInfo
- 8.429.2.16 **BYTE\*** nasGetSysInfoResp::pLTEVoiceSupportSysInfo
- 8.429.2.17 **CallBarringSysInfo\*** nasGetSysInfoResp::pWCDMACallBarringSysInfo
- 8.429.2.18 **BYTE\*** nasGetSysInfoResp::pWCDMACipherDomainSysInfo
- 8.429.2.19 **GSMSrvStatusInfo\*** nasGetSysInfoResp::pWCDMASrvStatusInfo
- 8.429.2.20 **WCDMASysInfo\*** nasGetSysInfoResp::pWCDMASysInfo

## 8.430 nasGetTxRxInfoReq Struct Reference

### Data Fields

- [BYTE radio\\_if](#)

### 8.430.1 Detailed Description

This structure contains the GetTxRxInfoReq request parameters

#### Parameters

<i>radio_if</i>	[Mandatory] <ul style="list-style-type: none"> <li>• Radio interface technology of the signal being measured</li> <li>• Valid Values               <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - CDMA</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1XEVD0 - HDR</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> </ul> </li> </ul>
-----------------	---

### 8.430.2 Field Documentation

- 8.430.2.1 **BYTE** nasGetTxRxInfoReq::radio\_if

## 8.431 nasGetTxRxInfoResp Struct Reference

### Data Fields

- [rxInfo](#) \* [pRXChain0Info](#)
- [rxInfo](#) \* [pRXChain1Info](#)
- [txInfo](#) \* [pTXInfo](#)

### 8.431.1 Detailed Description

This structure contains the GetTxRxInfoResp response parameters.

#### Parameters

<i>pRXChain0Info</i>	[Optional] • See <a href="#">rxInfo</a> for more information.
<i>pRXChain1Info</i>	[Optional] • See <a href="#">rxInfo</a> for more information.
<i>pTXInfo</i>	[Optional] • See <a href="#">txInfo</a> for more information.

### 8.431.2 Field Documentation

8.431.2.1 [rxInfo](#)\* [nasGetTxRxInfoResp::pRXChain0Info](#)

8.431.2.2 [rxInfo](#)\* [nasGetTxRxInfoResp::pRXChain1Info](#)

8.431.2.3 [txInfo](#)\* [nasGetTxRxInfoResp::pTXInfo](#)

## 8.432 NASGWAcqOrderPrefTlv Struct Reference

### Data Fields

- [uint8\\_t](#) [TlvPresent](#)
- [uint32\\_t](#) [GWAcqOrderPref](#)

### 8.432.1 Field Documentation

8.432.1.1 [uint32\\_t](#) [NASGWAcqOrderPrefTlv::GWAcqOrderPref](#)

8.432.1.2 [uint8\\_t](#) [NASGWAcqOrderPrefTlv::TlvPresent](#)

## 8.433 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](#)

### 8.433.1 Detailed Description

This structure contains the SLQSNasIndicationRegisterExt request parameters.

#### Parameters

<i>pSystem-SelectionInd</i>	[Optional] <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 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pDDTMInd</i>	[Optional] <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.  <a href="#">tFNDDTM</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pServing-SystemInd</i>	[Optional] <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 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pDualStandBy-PrefInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Dual Standby Preference indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNDualStandByPref</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSubscription-InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Subscription Information indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNSubscriptionInfo</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pNetworkTimeInd</i>	[Optional] <ul style="list-style-type: none"> <li>Network Time indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNNetworkTime</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pSysInfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>System Information indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNSysInfo</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pSignalStrengthInd</i>	[Optional] <ul style="list-style-type: none"> <li>Signal Strength indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNSigInfo</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pErrorRateInd</i>	[Optional] <ul style="list-style-type: none"> <li>Error Rate indication registration. The following callbacks would not be invoked if the indication is disabled.  <a href="#">tFNErrRate</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pHDRNewUATI-AssInd</i>	[Optional] <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.  <a href="#">tFNHDRUATIUpdate</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </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.  <a href="#">tFNHDRSessionClose</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </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.  <a href="#">tFNManagedRoaming</a> <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>

**Note**

At least one parameter must be provided as request. 'NULL' value confirms that the indication value is not sent.

## 8.433.2 Field Documentation

- 8.433.2.1 **BYTE\*** `nasIndicationRegisterReq::pDDTMInd`
- 8.433.2.2 **BYTE\*** `nasIndicationRegisterReq::pDualStandByPrefInd`
- 8.433.2.3 **BYTE\*** `nasIndicationRegisterReq::pErrorRateInd`
- 8.433.2.4 **BYTE\*** `nasIndicationRegisterReq::pHDRNewUATIAssInd`
- 8.433.2.5 **BYTE\*** `nasIndicationRegisterReq::pHDRSessionCloseInd`
- 8.433.2.6 **BYTE\*** `nasIndicationRegisterReq::pLTECphyCa`
- 8.433.2.7 **BYTE\*** `nasIndicationRegisterReq::pManagedRoamingInd`
- 8.433.2.8 **BYTE\*** `nasIndicationRegisterReq::pNetworkTimeInd`
- 8.433.2.9 **BYTE\*** `nasIndicationRegisterReq::pServingSystemInd`
- 8.433.2.10 **BYTE\*** `nasIndicationRegisterReq::pSignalStrengthInd`
- 8.433.2.11 **BYTE\*** `nasIndicationRegisterReq::pSubscriptionInfoInd`
- 8.433.2.12 **BYTE\*** `nasIndicationRegisterReq::pSysInfoInd`
- 8.433.2.13 **BYTE\*** `nasIndicationRegisterReq::pSystemSelectionInd`

## 8.434 nasInitNetworkReg Struct Reference

### Data Fields

- [ULONG](#) `regAction`
- [MNRInfo](#) \* `pMNRInfo`
- [ULONG](#) \* `pChangeDuration`
- [BOOL](#) \* `pMncPcsDigitStatus`

### 8.434.1 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>pMNRInfo</code> must also be included for the command to process successfully and supported only for 3GPP.</li> </ul> </li> </ul>
<i>pMNRInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">MNRInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">MNRInfo</a> for more information</li> </ul> </li> </ul>

<i>pChangeDuration</i>	[Optional] <ul style="list-style-type: none"> <li>Duration of the change.             <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 the client</li> </ul> </li> </ul>
<i>pMncPcsDigitStatus</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>

## 8.434.2 Field Documentation

8.434.2.1 **ULONG\*** nasInitNetworkReg::pChangeDuration

8.434.2.2 **BOOL\*** nasInitNetworkReg::pMncPcsDigitStatus

8.434.2.3 **MNRInfo\*** nasInitNetworkReg::pMNRInfo

8.434.2.4 **ULONG** nasInitNetworkReg::regAction

## 8.435 NASLTEBandPreferenceTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [LTEBandPref](#)

## 8.435.1 Field Documentation

8.435.1.1 **uint64\_t** NASLTEBandPreferenceTlv::LTEBandPref

8.435.1.2 **uint8\_t** NASLTEBandPreferenceTlv::TlvPresent

## 8.436 NASLteNasReleaseInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [nas\\_release](#)
- uint8\_t [nas\\_major](#)
- uint8\_t [nas\\_minor](#)

## 8.436.1 Field Documentation

8.436.1.1 **uint8\_t** NASLteNasReleaseInfoTlv::nas\_major

8.436.1.2 **uint8\_t** NASLteNasReleaseInfoTlv::nas\_minor

8.436.1.3 **uint8\_t** NASLteNasReleaseInfoTlv::nas\_release

8.436.1.4 `uint8_t NASLteNasReleaseInfoTlv::TlvPresent`

## 8.437 NASModePreferenceTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint16_t ModePref`

### 8.437.1 Field Documentation

8.437.1.1 `uint16_t NASModePreferenceTlv::ModePref`

8.437.1.2 `uint8_t NASModePreferenceTlv::TlvPresent`

## 8.438 NASNetSelPreferenceTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t NetSelPref`

### 8.438.1 Field Documentation

8.438.1.1 `uint8_t NASNetSelPreferenceTlv::NetSelPref`

8.438.1.2 `uint8_t NASNetSelPreferenceTlv::TlvPresent`

## 8.439 nasNetworkTime Struct Reference

### Data Fields

- `UniversalTime universalTime`
- `BYTE * pTimeZone`
- `BYTE * pDayltSavAdj`
- `BYTE * pRadioInterface`

### 8.439.1 Detailed Description

Structure for storing the `nasSysInfo` indication parameters.

#### Parameters

<i>universalTime</i>	<ul style="list-style-type: none"> <li>• See <code>UniversalTime</code> 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>
<i>pRadioInterface</i>	<ul style="list-style-type: none"> <li>Radio interface from which the information comes</li> <li>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>

## 8.439.2 Field Documentation

8.439.2.1 **BYTE\*** nasNetworkTime::pDayltSavAdj

8.439.2.2 **BYTE\*** nasNetworkTime::pRadioInterface

8.439.2.3 **BYTE\*** nasNetworkTime::pTimeZone

8.439.2.4 **UniversalTime** nasNetworkTime::universalTime

## 8.440 nasOperatorNameResp Struct Reference

### Data Fields

- [serviceName](#) \* [pSvcProviderName](#)
- [operatorPLMNList](#) \* [pOperatorPLMNList](#)
- [PLMNNetworkName](#) \* [pPLMNNetworkName](#)
- [operatorNameString](#) \* [pOperatorNameString](#)
- [PLMNNetworkNameData](#) \* [pNITZInformation](#)

### 8.440.1 Detailed Description

This structure contains Operator Name Data related from multiple sources.

#### Parameters

<i>pSvcProviderName</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">serviceName</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> </ul>
<i>pOperatorPLMNList</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>

### 8.440.2 Field Documentation

8.440.2.1 **PLMNNetworkNameData\*** nasOperatorNameResp::pNITZInformation

8.440.2.2 **operatorNameString\*** nasOperatorNameResp::pOperatorNameString

8.440.2.3 **operatorPLMNList\*** nasOperatorNameResp::pOperatorPLMNList

8.440.2.4 **PLMNNetworkName\*** nasOperatorNameResp::pPLMNNetworkName

8.440.2.5 **serviceProviderName\*** nasOperatorNameResp::pSvcProviderName

## 8.441 NASOTAMessageTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [message\\_type](#)
- uint16\_t [data\\_len](#)
- uint8\_t [data\\_buf](#) [2048]

### 8.441.1 Field Documentation

8.441.1.1 uint8\_t NASOTAMessageTlv::data\_buf[2048]

8.441.1.2 uint16\_t NASOTAMessageTlv::data\_len

8.441.1.3 uint32\_t NASOTAMessageTlv::message\_type

8.441.1.4 uint8\_t NASOTAMessageTlv::TlvPresent

## 8.442 NASPhyCaAggPcellInfo Struct Reference

### Data Fields

- uint32\_t [pci](#)
- uint32\_t [freq](#)
- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) dl\_bw\_value

- uint32\_t [iLTEbandValue](#)
- uint8\_t [TlvPresent](#)

### 8.442.1 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>

### 8.442.2 Field Documentation

8.442.2.1 [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) NASPhyCaAggPcellInfo::dl\_bw\_value

8.442.2.2 uint32\_t NASPhyCaAggPcellInfo::freq

8.442.2.3 uint32\_t NASPhyCaAggPcellInfo::iLTEbandValue

8.442.2.4 uint32\_t NASPhyCaAggPcellInfo::pci

8.442.2.5 uint8\_t NASPhyCaAggPcellInfo::TlvPresent

## 8.443 NASPhyCaAggScellIDBw Struct Reference

#### Data Fields

- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) dl\_bw\_value
- uint8\_t [TlvPresent](#)

### 8.443.1 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>
--------------------	--

## 8.443.2 Field Documentation

8.443.2.1 `LIBPACK_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellIDIBw::dl_bw_value`8.443.2.2 `uint8_t NASPhyCaAggScellIDIBw::TlvPresent`

## 8.444 NASPhyCaAggScellIndex Struct Reference

## Data Fields

- `uint8_t` [scell\\_idx](#)
- `uint8_t` [TlvPresent](#)

## 8.444.1 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>

## 8.444.2 Field Documentation

8.444.2.1 `uint8_t NASPhyCaAggScellIndex::scell_idx`8.444.2.2 `uint8_t NASPhyCaAggScellIndex::TlvPresent`

## 8.445 NASPhyCaAggScellIndType Struct Reference

## Data Fields

- `uint32_t` [pci](#)
- `uint32_t` [freq](#)
- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) [scell\\_state](#)
- `uint8_t` [TlvPresent](#)

## 8.445.1 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>

## 8.445.2 Field Documentation

8.445.2.1 uint32\_t NASPhyCaAggScellIndType::freq

8.445.2.2 uint32\_t NASPhyCaAggScellIndType::pci

8.445.2.3 LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE NASPhyCaAggScellIndType::scell\_state

8.445.2.4 uint8\_t NASPhyCaAggScellIndType::TlvPresent

## 8.446 NASPhyCaAggScellInfo Struct Reference

## Data Fields

- uint32\_t [pci](#)
- uint32\_t [freq](#)
- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) dl\_bw\_value
- uint32\_t [iLTEbandValue](#)
- [LIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) scell\_state
- uint8\_t [TlvPresent](#)

## 8.446.1 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>

## 8.446.2 Field Documentation

8.446.2.1 **LIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB** NASPhyCaAggScellInfo::dl\_bw\_value

8.446.2.2 **uint32\_t** NASPhyCaAggScellInfo::freq

8.446.2.3 **uint32\_t** NASPhyCaAggScellInfo::iLTEbandValue

8.446.2.4 **uint32\_t** NASPhyCaAggScellInfo::pci

8.446.2.5 **LIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE** NASPhyCaAggScellInfo::scell\_state

8.446.2.6 **uint8\_t** NASPhyCaAggScellInfo::TlvPresent

## 8.447 nasPLMNNameReq Struct Reference

### Data Fields

- [WORD](#) mcc
- [WORD](#) mnc
- [BYTE](#) \* pMncPcsStatus

### 8.447.1 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>

<i>pMncPcsStatus</i>	<ul style="list-style-type: none"><li>• MNC PCS Digit Include Status</li><li>• Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01).</li><li>• Values<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></li></ul>
----------------------	--

**Note**

If pMncPcsStatus is not present, an MNC smaller than 100 is assumed to be a two-digit value, and an MNC greater than or equal to 100 is assumed to be a three digit value.

**8.447.2 Field Documentation**

8.447.2.1 **WORD** nasPLMNNameReq::mcc

8.447.2.2 **WORD** nasPLMNNameReq::mnc

8.447.2.3 **BYTE\*** nasPLMNNameReq::pMncPcsStatus

**8.448 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 \[255\]](#)

**8.448.1 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

## 8.448.2 Field Documentation

8.448.2.1 BYTE nasPLMNNameResp::longName[255]

8.448.2.2 BYTE nasPLMNNameResp::longNameCI

8.448.2.3 BYTE nasPLMNNameResp::longNameEn

8.448.2.4 BYTE nasPLMNNameResp::longNameLen

8.448.2.5 BYTE nasPLMNNameResp::longNameSB

8.448.2.6 BYTE nasPLMNNameResp::shortName[255]

8.448.2.7 BYTE nasPLMNNameResp::shortNameCI

8.448.2.8 BYTE nasPLMNNameResp::shortNameEn

8.448.2.9 BYTE nasPLMNNameResp::shortNameLen

8.448.2.10 BYTE nasPLMNNameResp::shortNameSB

8.448.2.11 BYTE nasPLMNNameResp::spn[255]

8.448.2.12 BYTE nasPLMNNameResp::spnEncoding

8.448.2.13 BYTE nasPLMNNameResp::spnLength

## 8.449 NASPRLPreferenceTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [PRLPref](#)

### 8.449.1 Field Documentation

8.449.1.1 uint16\_t NASPRLPreferenceTlv::PRLPref

8.449.1.2 uint8\_t NASPRLPreferenceTlv::TlvPresent

## 8.450 NASQmiCbkJnasSwtOTAMessageInd Struct Reference

## Data Fields

- [NASOTAMessageTlv otaMsgTlv](#)
- [NASLteNasReleaseInfoTlv nasRelInfoTlv](#)
- [NASTimeInfoTlv timeTlv](#)

### 8.450.1 Field Documentation

8.450.1.1 [NASLteNasReleaseInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::nasRelInfoTlv

8.450.1.2 [NASOTAMessageTlv](#) NASQmiCbkJnasSwtOTAMessageInd::otaMsgTlv

8.450.1.3 [NASTimeInfoTlv](#) NASQmiCbkJnasSwtOTAMessageInd::timeTlv

## 8.451 NASQmiCbkJnasSystemSelPrefInd Struct Reference

## Data Fields

- [NASEmergencyModeTlv EMTlv](#)
- [NASModePreferenceTlv MPTlv](#)
- [NASBandPreferenceTlv BPTlv](#)
- [NASPRLPreferenceTlv PRLPTlv](#)
- [NASRoamPreferenceTlv RPTlv](#)
- [NASLTEBandPreferenceTlv LBPTlv](#)
- [NASNetSelPreferenceTlv NSPTlv](#)
- [NASServDomainPrefTlv SDPTlv](#)
- [NASGWAcqOrderPrefTlv GWAOPTlv](#)

### 8.451.1 Field Documentation

8.451.1.1 [NASBandPreferenceTlv](#) NASQmiCbkJnasSystemSelPrefInd::BPTlv

8.451.1.2 [NASEmergencyModeTlv](#) NASQmiCbkJnasSystemSelPrefInd::EMTlv

8.451.1.3 [NASGWAcqOrderPrefTlv](#) NASQmiCbkJnasSystemSelPrefInd::GWAOPTlv

8.451.1.4 [NASLTEBandPreferenceTlv](#) NASQmiCbkJnasSystemSelPrefInd::LBPTlv

8.451.1.5 [NASModePreferenceTlv](#) NASQmiCbkJnasSystemSelPrefInd::MPTlv

8.451.1.6 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.451.1.7 **NASPRLPReferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.451.1.8 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv

8.451.1.9 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

## 8.452 NASRoamPreferenceTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [RoamPref](#)

### 8.452.1 Field Documentation

8.452.1.1 uint16\_t NASRoamPreferenceTlv::RoamPref

8.452.1.2 uint8\_t NASRoamPreferenceTlv::TlvPresent

## 8.453 NASServDomainPrefTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [SrvDomainPref](#)

### 8.453.1 Field Documentation

8.453.1.1 uint32\_t NASServDomainPrefTlv::SrvDomainPref

8.453.1.2 uint8\_t NASServDomainPrefTlv::TlvPresent

## 8.454 NASServingSystemInfo Struct Reference

### Data Fields

- uint8\_t [registrationState](#)
- uint8\_t [csAttachState](#)
- uint8\_t [psAttachState](#)
- uint8\_t [selectedNetwork](#)
- uint8\_t [radioInterfaceNo](#)
- uint8\_t [radioInterfaceList](#) [255]
- uint8\_t [hdrPersonality](#)

### 8.454.1 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

## 8.454.2 Field Documentation

8.454.2.1 `uint8_t NAServingSystemInfo::csAttachState`

8.454.2.2 `uint8_t NAServingSystemInfo::hdrPersonality`

8.454.2.3 `uint8_t NAServingSystemInfo::psAttachState`

8.454.2.4 uint8\_t NAServingSystemInfo::radioInterfaceList[255]

8.454.2.5 uint8\_t NAServingSystemInfo::radioInterfaceNo

8.454.2.6 uint8\_t NAServingSystemInfo::registrationState

8.454.2.7 uint8\_t NAServingSystemInfo::selectedNetwork

## 8.455 nasSigInfo Struct Reference

### Data Fields

- [CDMASSInfo](#) \* [pCDMASigInfo](#)
- [HDRSSInfo](#) \* [pHDRSigInfo](#)
- [INT8](#) \* [pGSMSigInfo](#)
- [CDMASSInfo](#) \* [pWCDMASigInfo](#)
- [LTESSInfo](#) \* [pLTESigInfo](#)
- [INT8](#) \* [pRscp](#)
- [TDSCDMASigInfoExt](#) \* [pTDSCDMASigInfoExt](#)

### 8.455.1 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>

### 8.455.2 Field Documentation

8.455.2.1 [CDMASSInfo](#)\* [nasSigInfo](#)::[pCDMASigInfo](#)

8.455.2.2 INT8\* nasSigInfo::pGMSigInfo

8.455.2.3 HDRSSInfo\* nasSigInfo::pHDRSigInfo

8.455.2.4 LTESInfo\* nasSigInfo::pLTESigInfo

8.455.2.5 INT8\* nasSigInfo::pRscp

8.455.2.6 TDSCDMASigInfoExt\* nasSigInfo::pTDSCDMASigInfoExt

8.455.2.7 CDMASInfo\* nasSigInfo::pWCDMASigInfo

## 8.456 nasSwiGetChannelLockResp Struct Reference

### Data Fields

- [wcdmaUARFCN](#) \* [pWcdmaUARFCN](#)
- [lteEARFCN](#) \* [pLteEARFCN](#)
- [ltePCI](#) \* [pLtePCI](#)

### 8.456.1 Detailed Description

This structure contains the SLQSNASSwiGetChannelLock response parameters.

#### Parameters

<i>pWcdmaUARFCN</i>	[Optional] • See <a href="#">wcdmaUARFCN</a> for more information
<i>pLteEARFCN</i>	[Optional] • See <a href="#">lteEARFCN</a> for more information
<i>pLtePCI</i>	[Optional] • See <a href="#">ltePCI</a> for more information

### 8.456.2 Field Documentation

8.456.2.1 lteEARFCN\* nasSwiGetChannelLockResp::pLteEARFCN

8.456.2.2 ltePCI\* nasSwiGetChannelLockResp::pLtePCI

8.456.2.3 wcdmaUARFCN\* nasSwiGetChannelLockResp::pWcdmaUARFCN

## 8.457 NasSwiIndReg Struct Reference

### Data Fields

- [BYTE](#) lteEsmUI
- [BYTE](#) lteEsmDI
- [BYTE](#) lteEmmUI
- [BYTE](#) lteEmmDI
- [BYTE](#) gsmUmtsUI

- [BYTE gsmUmtsDI](#)
- [BYTE \\* pRankIndicatorInd](#)

### 8.457.1 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>
<i>pRankIndicator-Ind</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report Rank Indicator messages</li> </ul>

### 8.457.2 Field Documentation

8.457.2.1 **BYTE** NasSwlIndReg::gsmUmtsDI

8.457.2.2 **BYTE** NasSwlIndReg::gsmUmtsUI

8.457.2.3 **BYTE** NasSwlIndReg::lteEmmDI

8.457.2.4 **BYTE** NasSwlIndReg::lteEmmUI

8.457.2.5 **BYTE** NasSwlIndReg::lteEsmDI

8.457.2.6 **BYTE** NasSwlIndReg::lteEsmUI

8.457.2.7 **BYTE\*** NasSwlIndReg::pRankIndicatorInd

## 8.458 nasSwtSetChannelLockReq Struct Reference

### Data Fields

- [wcdmaUARFCN](#) \* [pWcdmaUARFCN](#)
- [lteEARFCN](#) \* [pLteEARFCN](#)
- [ltePCI](#) \* [pLtePCI](#)

### 8.458.1 Detailed Description

This structure contains the SLQSNASSwtSetChannelLock response parameters.

#### Parameters

<i>pWcdmaUARFCN</i>	[Optional] • See <a href="#">wcdmaUARFCN</a> for more information
<i>pLteEARFCN</i>	[Optional] • See <a href="#">lteEARFCN</a> for more information
<i>pLtePCI</i>	[Optional] • See <a href="#">ltePCI</a> for more information

### 8.458.2 Field Documentation

8.458.2.1 [lteEARFCN](#)\* [nasSwtSetChannelLockReq::pLteEARFCN](#)

8.458.2.2 [ltePCI](#)\* [nasSwtSetChannelLockReq::pLtePCI](#)

8.458.2.3 [wcdmaUARFCN](#)\* [nasSwtSetChannelLockReq::pWcdmaUARFCN](#)

## 8.459 nasSysInfo Struct Reference

### Data Fields

- [SrvStatusInfo](#) \* [pCDMASrvStatusInfo](#)
- [SrvStatusInfo](#) \* [pHDRSrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pGSMSrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pWCDMASrvStatusInfo](#)
- [GSMSrvStatusInfo](#) \* [pLTERsrvStatusInfo](#)
- [CDMASysInfo](#) \* [pCDMASysInfo](#)
- [HDRSysInfo](#) \* [pHDRSysInfo](#)
- [GSMSysInfo](#) \* [pGSMSysInfo](#)
- [WCDMASysInfo](#) \* [pWCDMASysInfo](#)
- [LTERSysInfo](#) \* [pLTERSysInfo](#)
- [AddCDMASysInfo](#) \* [pAddCDMASysInfo](#)
- [WORD](#) \* [pAddHDRSysInfo](#)
- [AddSysInfo](#) \* [pAddGSMSysInfo](#)
- [AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [WORD](#) \* [pAddLTERSysInfo](#)
- [CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)

- [BYTE \\* pLTEVoiceSupportSysInfo](#)
- [BYTE \\* pGSMCipherDomainSysInfo](#)
- [BYTE \\* pWCDMACipherDomainSysInfo](#)
- [BYTE \\* pSysInfoNoChange](#)

### 8.459.1 Detailed Description

Structure for storing the [nasSysInfo](#) indication 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>pGSMCallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pWCDMACallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pLTEVoiceSupportSysInfo</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>pGSMCipherDomainSysInfo</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>pWCDMACipherDomainSysInfo</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>pSysInfoNoChange</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>

## 8.459.2 Field Documentation

8.459.2.1 **AddCDMASysInfo\*** nasSysInfo::pAddCDMASysInfo

8.459.2.2 **AddSysInfo\*** nasSysInfo::pAddGSM SysInfo

8.459.2.3 **WORD\*** nasSysInfo::pAddHDR SysInfo

8.459.2.4 **WORD\*** nasSysInfo::pAddLTESysInfo

8.459.2.5 **AddSysInfo\*** nasSysInfo::pAddWCDMASysInfo

8.459.2.6 **SrvStatusInfo\*** nasSysInfo::pCDMASrvStatusInfo

8.459.2.7 **CDMASysInfo\*** nasSysInfo::pCDMASysInfo

- 8.459.2.8 **CallBarringSysInfo\*** nasSysInfo::pGSMCallBarringSysInfo
- 8.459.2.9 **BYTE\*** nasSysInfo::pGSMCipherDomainSysInfo
- 8.459.2.10 **GSMSrvStatusInfo\*** nasSysInfo::pGSMSrvStatusInfo
- 8.459.2.11 **GSMSysInfo\*** nasSysInfo::pGSMSysInfo
- 8.459.2.12 **SrvStatusInfo\*** nasSysInfo::pHDSrvStatusInfo
- 8.459.2.13 **HDRSysInfo\*** nasSysInfo::pHDSysInfo
- 8.459.2.14 **GSMSrvStatusInfo\*** nasSysInfo::pLTERsrvStatusInfo
- 8.459.2.15 **LTESysInfo\*** nasSysInfo::pLTESysInfo
- 8.459.2.16 **BYTE\*** nasSysInfo::pLTEVoiceSupportSysInfo
- 8.459.2.17 **BYTE\*** nasSysInfo::pSysInfoNoChange
- 8.459.2.18 **CallBarringSysInfo\*** nasSysInfo::pWCDMACallBarringSysInfo
- 8.459.2.19 **BYTE\*** nasSysInfo::pWCDMACipherDomainSysInfo
- 8.459.2.20 **GSMSrvStatusInfo\*** nasSysInfo::pWCDMASrvStatusInfo
- 8.459.2.21 **WCDMASysInfo\*** nasSysInfo::pWCDMASysInfo

## 8.460 NASTimeInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [time](#)

### 8.460.1 Field Documentation

- 8.460.1.1 uint64\_t NASTimeInfoTlv::time
- 8.460.1.2 uint8\_t NASTimeInfoTlv::TlvPresent

## 8.461 netSelectionPref Struct Reference

### Data Fields

- [BYTE](#) netReg
- [WORD](#) mcc
- [WORD](#) mnc

### 8.461.1 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>

## 8.461.2 Field Documentation

8.461.2.1 WORD netSelectionPref::mcc

8.461.2.2 WORD netSelectionPref::mnc

8.461.2.3 BYTE netSelectionPref::netReg

## 8.462 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

## 8.462.1 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>

## 8.462.2 Field Documentation

8.462.2.1 **ULONGLONG** NetStats::rx\_bytes

8.462.2.2 **ULONG** NetStats::rx\_errors

8.462.2.3 **ULONG** NetStats::rx\_overflows

8.462.2.4 **ULONG** NetStats::rx\_packets

8.462.2.5 **ULONGLONG** NetStats::tx\_bytes

8.462.2.6 **ULONG** NetStats::tx\_errors

8.462.2.7 **ULONG** NetStats::tx\_overflows

8.462.2.8 **ULONG** NetStats::tx\_packets

## 8.463 NetworkDebugResp Struct Reference

### Data Fields

- [BYTE](#) \* [pObjectVer](#)
- [NetworkStat1x](#) \* [pNetworkStat1x](#)
- [NetworkStatEVDO](#) \* [pNetworkStatEVDO](#)
- [DeviceConfigDetail](#) \* [pDeviceConfigDetail](#)
- [DataStatusDetail](#) \* [pDataStatusDetail](#)

### 8.463.1 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>

## 8.463.2 Field Documentation

8.463.2.1 **DataStatusDetail\*** NetworkDebugResp::pDataStatusDetail

8.463.2.2 **DeviceConfigDetail\*** NetworkDebugResp::pDeviceConfigDetail

8.463.2.3 **NetworkStat1x\*** NetworkDebugResp::pNetworkStat1x

8.463.2.4 **NetworkStatEVDO\*** NetworkDebugResp::pNetworkStatEVDO

8.463.2.5 **BYTE\*** NetworkDebugResp::pObjectVer

## 8.464 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](#)

### 8.464.1 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>

## 8.464.2 Field Documentation

8.464.2.1 **BYTE** NetworkStat1x::ActSetCnt

8.464.2.2 **BYTE** NetworkStat1x::NeighborSetCnt

8.464.2.3 **ActPilotPNElement\*** NetworkStat1x::pActPilotPNElements

8.464.2.4 **WORD\*** NetworkStat1x::pNeighborSetPilotPN

8.464.2.5 **WORD** NetworkStat1x::RX\_EC\_IO

8.464.2.6 **ULONG** NetworkStat1x::RX\_PWR

8.464.2.7 **WORD** NetworkStat1x::SO

8.464.2.8 **BYTE** NetworkStat1x::State

8.464.2.9 **ULONG** NetworkStat1x::TX\_PWR

## 8.465 NetworkStatEVDO Struct Reference

### Data Fields

- [BYTE](#) State
- [BYTE](#) MACIndex
- [BYTE](#) SectorIDLen
- [WORD](#) \* pSectorID
- [WORD](#) RX\_PWR
- [WORD](#) PER
- [WORD](#) PilotEnergy
- [BYTE](#) SNR

### 8.465.1 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>

## 8.465.2 Field Documentation

- 8.465.2.1 **BYTE** NetworkStatEVDO::MACIndex
- 8.465.2.2 **WORD** NetworkStatEVDO::PER
- 8.465.2.3 **WORD** NetworkStatEVDO::PilotEnergy
- 8.465.2.4 **WORD\*** NetworkStatEVDO::pSectorID
- 8.465.2.5 **WORD** NetworkStatEVDO::RX\_PWR
- 8.465.2.6 **BYTE** NetworkStatEVDO::SectorIDLen
- 8.465.2.7 **BYTE** NetworkStatEVDO::SNR
- 8.465.2.8 **BYTE** NetworkStatEVDO::State

## 8.466 newMTMessageTlv Struct Reference

### Data Fields

- [uint8\\_t TlvPresent](#)
- [sMSMTMessageInfo MTMessageInfo](#)

### 8.466.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>MTMessageInfo</i>	<ul style="list-style-type: none"> <li>• MT Message</li> <li>• See <a href="#">sMSMTMessageInfo</a> for more information</li> </ul>

### 8.466.2 Field Documentation

- 8.466.2.1 **sMSMTMessageInfo** newMTMessageTlv::MTMessageInfo
- 8.466.2.2 **uint8\_t** newMTMessageTlv::TlvPresent

## 8.467 newPwdData Struct Reference

### Data Fields

- [BYTE newPwd](#) [4]
- [BYTE newPwdAgain](#) [4]

### 8.467.1 Detailed Description

This structure contains New Password Data.

## Parameters

<i>newPwd</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>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>

## 8.467.2 Field Documentation

8.467.2.1 BYTE newPwdData::newPwd[4]

8.467.2.2 BYTE newPwdData::newPwdAgain[4]

## 8.468 nmrCellInfo Struct Reference

## Data Fields

- [ULONG nmrCellID](#)
- [BYTE nmrPlmn](#) [3]
- [WORD nmrLac](#)
- [WORD nmrArfcn](#)
- [BYTE nmrBsic](#)
- [WORD nmrRxLev](#)

## 8.468.1 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>

## 8.468.2 Field Documentation

8.468.2.1 WORD nmrCellInfo::nmrArfcn

8.468.2.2 BYTE nmrCellInfo::nmrBsic

8.468.2.3 ULONG nmrCellInfo::nmrCellID

8.468.2.4 WORD nmrCellInfo::nmrLac

8.468.2.5 BYTE nmrCellInfo::nmrPlmn[3]

8.468.2.6 WORD nmrCellInfo::nmrRxLev

## 8.469 NSSAudioCtrl Struct Reference

### Data Fields

- BYTE [upLink](#)
- BYTE [downLink](#)

### 8.469.1 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>

## 8.469.2 Field Documentation

8.469.2.1 BYTE NSSAudioCtrl::downLink

8.469.2.2 BYTE NSSAudioCtrl::upLink

## 8.470 NWProfile Struct Reference

### Data Fields

- WORD tech
- BYTE \* pProfSz
- WORD \* pProfValues

### 8.470.1 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>	

## 8.470.2 Field Documentation

8.470.2.1 BYTE\* NWProfile::pProfSz

8.470.2.2 WORD\* NWProfile::pProfValues

8.470.2.3 WORD NWProfile::tech

## 8.471 omaDmConfigTlv Struct Reference

### Data Fields

- BYTE state
- BYTE userInputReq
- USHORT userInputTimeout
- USHORT alertmsglength
- BYTE alertmsg [256]

### 8.471.1 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>

## 8.471.2 Field Documentation

8.471.2.1 **BYTE** omaDmConfigTlv::alertmsg[256]

8.471.2.2 **USHORT** omaDmConfigTlv::alertmsglength

8.471.2.3 **BYTE** omaDmConfigTlv::state

8.471.2.4 **BYTE** omaDmConfigTlv::userInputReq

8.471.2.5 **USHORT** omaDmConfigTlv::userInputTimeout

## 8.472 omaDmConfigTlvExt Struct Reference

### Data Fields

- [BYTE](#) state
- [BYTE](#) userInputReq
- [USHORT](#) userInputTimeout
- [USHORT](#) alertmsglength
- [BYTE](#) alertmsg [256]

### 8.472.1 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>	- 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".
<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>
-----------------	--

## 8.472.2 Field Documentation

8.472.2.1 **BYTE** omaDmConfigTlvExt::alertmsg[256]

8.472.2.2 **USHORT** omaDmConfigTlvExt::alertmsglength

8.472.2.3 **BYTE** omaDmConfigTlvExt::state

8.472.2.4 **BYTE** omaDmConfigTlvExt::userInputReq

8.472.2.5 **USHORT** omaDmConfigTlvExt::userInputTimeout

## 8.473 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](#)

### 8.473.1 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>

## 8.473.2 Field Documentation

8.473.2.1 **BYTE** omaDmFotaTlv::description[256]

8.473.2.2 **USHORT** omaDmFotaTlv::descriptionlength

8.473.2.3 **ULONG** omaDmFotaTlv::fwdloadsize

- 8.473.2.4    **ULONG** omaDmFotaTlv::fwloadComplete
- 8.473.2.5    **USHORT** omaDmFotaTlv::namelength
- 8.473.2.6    **BYTE** omaDmFotaTlv::package\_name[256]
- 8.473.2.7    **BYTE** omaDmFotaTlv::sessionType
- 8.473.2.8    **BYTE** omaDmFotaTlv::severity
- 8.473.2.9    **BYTE** omaDmFotaTlv::state
- 8.473.2.10   **USHORT** omaDmFotaTlv::updateCompleteStatus
- 8.473.2.11   **BYTE** omaDmFotaTlv::userInputReq
- 8.473.2.12   **USHORT** omaDmFotaTlv::userInputTimeout
- 8.473.2.13   **BYTE** omaDmFotaTlv::version[256]
- 8.473.2.14   **USHORT** omaDmFotaTlv::versionlength

## 8.474    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]

### 8.474.1   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>
--------------------	--

## 8.474.2 Field Documentation

8.474.2.1 **BYTE** omaDmFotaTlvExt::description[256]

8.474.2.2 **USHORT** omaDmFotaTlvExt::descriptionlength

8.474.2.3 **USHORT** omaDmFotaTlvExt::fumoResultCode

8.474.2.4 **USHORT** omaDmFotaTlvExt::namelength

8.474.2.5 **BYTE** omaDmFotaTlvExt::package\_name[256]

8.474.2.6 **ULONG** omaDmFotaTlvExt::packageSize

8.474.2.7 **ULONG** omaDmFotaTlvExt::receivedBytes

8.474.2.8 **BYTE** omaDmFotaTlvExt::reserved

8.474.2.9 **BYTE** omaDmFotaTlvExt::state

8.474.2.10 **USHORT** omaDmFotaTlvExt::userInputTimeout

8.474.2.11 **BYTE** omaDmFotaTlvExt::version[256]

8.474.2.12 **USHORT** omaDmFotaTlvExt::versionlength

## 8.475 omaDmNotificationsTlv Struct Reference

### Data Fields

- [BYTE](#) notification
- [USHORT](#) sessionStatus

## 8.475.1 Field Documentation

8.475.1.1 **BYTE** omaDmNotificationsTlv::notification

8.475.1.2 **USHORT** omaDmNotificationsTlv::sessionStatus

## 8.476 operatorNameString Struct Reference

### Data Fields

- [BYTE](#) PLMNName [255]

### 8.476.1 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>
-----------------	---

### 8.476.2 Field Documentation

#### 8.476.2.1 BYTE operatorNameString::PLMNName[255]

## 8.477 OperatorPLMNData Struct Reference

### Data Fields

- [BYTE mcc](#) [3]
- [BYTE mnc](#) [3]
- [WORD lac1](#)
- [WORD lac2](#)
- [BYTE PLMNRecID](#)

### 8.477.1 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>

### 8.477.2 Field Documentation

#### 8.477.2.1 WORD OperatorPLMNData::lac1

8.477.2.2 WORD OperatorPLMNData::lac2

8.477.2.3 BYTE OperatorPLMNData::mcc[3]

8.477.2.4 BYTE OperatorPLMNData::mnc[3]

8.477.2.5 BYTE OperatorPLMNData::PLMNRecID

## 8.478 operatorPLMNList Struct Reference

### Data Fields

- [WORD numInstance](#)
- [OperatorPLMNData PLMNData](#) [255]

### 8.478.1 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>

### 8.478.2 Field Documentation

8.478.2.1 WORD operatorPLMNList::numInstance

8.478.2.2 OperatorPLMNData operatorPLMNList::PLMNData[255]

## 8.479 pack\_dms\_GetCustFeaturesV2\_t Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint8\_t [list\\_type](#)
- uint16\_t [Tlvresult](#)

### 8.479.1 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>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack Result</li> </ul>

## 8.479.2 Field Documentation

8.479.2.1 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::cust\_id[64+1]

8.479.2.2 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::list\_type

8.479.2.3 uint16\_t pack\_dms\_GetCustFeaturesV2\_t::Tlvresult

## 8.480 pack\_dms\_SetCrashAction\_t Struct Reference

### Data Fields

- uint8\_t [crashAction](#)

### 8.480.1 Detailed Description

Modem action in case of a crash

#### Parameters

<i>crashAction</i>	<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 DM port enumerated.</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</li> <li>2 - No Action</li> </ul>
--------------------	---

## 8.480.2 Field Documentation

8.480.2.1 uint8\_t pack\_dms\_SetCrashAction\_t::crashAction

## 8.481 pack\_dms\_SetCustFeature\_t Struct Reference

### Data Fields

- uint32\_t [GpsEnable](#)
- uint8\_t [DisableIMSI](#)
- uint16\_t [IPFamSupport](#)
- uint8\_t [RMAutoConnect](#)
- uint8\_t [GPSSel](#)
- uint8\_t [SMSSupport](#)
- uint8\_t [IsVoiceEnabled](#)
- uint8\_t [DHCPRelayEnabled](#)
- uint8\_t [GPSLPM](#)

### 8.481.1 Field Documentation

8.481.1.1 uint8\_t pack\_dms\_SetCustFeature\_t::DHCPRelayEnabled

8.481.1.2 uint8\_t pack\_dms\_SetCustFeature\_t::DisableIMSI

8.481.1.3 uint32\_t pack\_dms\_SetCustFeature\_t::GpsEnable

8.481.1.4 uint8\_t pack\_dms\_SetCustFeature\_t::GPSLPM

8.481.1.5 uint8\_t pack\_dms\_SetCustFeature\_t::GPSSel

8.481.1.6 uint16\_t pack\_dms\_SetCustFeature\_t::IPFamSupport

8.481.1.7 uint8\_t pack\_dms\_SetCustFeature\_t::IsVoiceEnabled

8.481.1.8 uint8\_t pack\_dms\_SetCustFeature\_t::RMAutoConnect

8.481.1.9 uint8\_t pack\_dms\_SetCustFeature\_t::SMSSupport

## 8.482 pack\_dms\_SetCustFeaturesV2\_t Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint16\_t [value\\_length](#)
- uint8\_t [cust\\_value](#) [8+1]
- uint16\_t [Tlvresult](#)

### 8.482.1 Detailed Description

This structure contains customization settings set to modem pack

#### 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 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>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack Result</li> </ul>

### 8.482.2 Field Documentation

8.482.2.1 uint8\_t pack\_dms\_SetCustFeaturesV2\_t::cust\_id[64+1]

8.482.2.2 uint8\_t pack\_dms\_SetCustFeaturesV2\_t::cust\_value[8+1]

8.482.2.3 uint16\_t pack\_dms\_SetCustFeaturesV2\_t::Tlvresult

8.482.2.4 uint16\_t pack\_dms\_SetCustFeaturesV2\_t::value\_length

## 8.483 pack\_dms\_SetEventReport\_t Struct Reference

### Data Fields

- uint8\_t [mode](#)

#### 8.483.1 Field Documentation

8.483.1.1 uint8\_t pack\_dms\_SetEventReport\_t::mode

## 8.484 pack\_dms\_SetPower\_t Struct Reference

### Data Fields

- uint32\_t [mode](#)
- uint16\_t [Tlvresult](#)

#### 8.484.1 Field Documentation

8.484.1.1 uint32\_t pack\_dms\_SetPower\_t::mode

8.484.1.2 uint16\_t pack\_dms\_SetPower\_t::Tlvresult

## 8.485 pack\_dms\_SetUSBComp\_t Struct Reference

### Data Fields

- uint8\_t [USBComp](#)
- uint16\_t [Tlvresult](#)

#### 8.485.1 Field Documentation

8.485.1.1 uint16\_t pack\_dms\_SetUSBComp\_t::Tlvresult

8.485.1.2 uint8\_t pack\_dms\_SetUSBComp\_t::USBComp

## 8.486 pack\_dms\_SLQSDmsSwilIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t [resetInfoInd](#)

#### 8.486.1 Detailed Description

## Parameters

<i>resetInfoInd</i> [IN]	<ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>0 - Disable</li> <li>1 - Enable</li> </ul> </li> </ul>
--------------------------	---

## 8.486.2 Field Documentation

8.486.2.1 uint8\_t pack\_dms\_SLQSDmsSwiIndicationRegister\_t::resetInfoInd

## 8.487 pack\_dms\_SLQSSwiSetDyingGaspCfg\_t Struct Reference

## Data Fields

- uint8\_t \* [pDestSMSNum](#)
- uint8\_t \* [pDestSMSContent](#)

## 8.487.1 Detailed Description

## Parameters

<i>pDestSMSNum</i> [IN]	<ul style="list-style-type: none"> <li>SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>Optional parameter.</li> </ul>
<i>pDestSMSContent</i> [IN]	<ul style="list-style-type: none"> <li>SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>Optional parameter.</li> </ul>

## 8.487.2 Field Documentation

8.487.2.1 uint8\_t\* pack\_dms\_SLQSSwiSetDyingGaspCfg\_t::pDestSMSContent

8.487.2.2 uint8\_t\* pack\_dms\_SLQSSwiSetDyingGaspCfg\_t::pDestSMSNum

## 8.488 pack\_dms\_UIMGetICCID\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

## 8.488.1 Detailed Description

This structure contains UIM Get ICCID pack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack result.</li> </ul>
------------------	--

### 8.488.2 Field Documentation

8.488.2.1 uint16\_t pack\_dms\_UIMGetlCCID\_t::Tlvresult

## 8.489 pack\_fms\_GetImagesPreference\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.489.1 Detailed Description

This structure contains the Get Image Preference information pack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Pack result</li></ul>
------------------	---

### 8.489.2 Field Documentation

8.489.2.1 uint16\_t pack\_fms\_GetImagesPreference\_t::Tlvresult

## 8.490 pack\_fms\_GetStoredImages\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.490.1 Detailed Description

This structure contains the Get Stored Images pack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Pack result</li></ul>
------------------	---

### 8.490.2 Field Documentation

8.490.2.1 uint16\_t pack\_fms\_GetStoredImages\_t::Tlvresult

## 8.491 pack\_fms\_SetImagesPreference\_t Struct Reference

### Data Fields

- uint32\_t [imageListSize](#)
- [FMSPrefImageList](#) \* [pImageList](#)
- uint32\_t [bForceDownload](#)
- uint8\_t [modemindex](#)

- uint16\_t [Tlvresult](#)

### 8.491.1 Detailed Description

This structure contains the Set Images Preference pack

#### Parameters

<i>imageListSize</i>	<ul style="list-style-type: none"> <li>• Image List Size</li> </ul>
<i>pImageList</i>	<ul style="list-style-type: none"> <li>• Image List</li> <li>• See <a href="#">FMSPrefImageList</a></li> </ul>
<i>bForceDownload</i>	<ul style="list-style-type: none"> <li>• 0 - Not Force Download.</li> <li>• 1 - Force Download.</li> </ul>
<i>modemindex</i>	<ul style="list-style-type: none"> <li>• Modem Index.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

### 8.491.2 Field Documentation

8.491.2.1 uint32\_t pack\_fms\_SetImagesPreference\_t::bForceDownload

8.491.2.2 uint32\_t pack\_fms\_SetImagesPreference\_t::imageListSize

8.491.2.3 uint8\_t pack\_fms\_SetImagesPreference\_t::modemindex

8.491.2.4 FMSPrefImageList\* pack\_fms\_SetImagesPreference\_t::pImageList

8.491.2.5 uint16\_t pack\_fms\_SetImagesPreference\_t::Tlvresult

## 8.492 pack\_loc\_Delete\_Assist\_Data\_t Struct Reference

### Data Fields

- loc\_SVInfo \* pSVInfo
- loc\_GnssData \* pGnssData
- loc\_CellDb \* pCellDb
- loc\_ClkInfo \* pClkInfo
- loc\_BdsSVInfo \* pBdsSVInfo
- uint16\_t [Tlvresult](#)

### 8.492.1 Detailed Description

This structure contains LOC delete assist data pack

## Parameters

<i>pSVInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">loc_SVInfo</a>. See <a href="#">loc_SVInfo</a> for more information</li> </ul>
<i>pGnssData</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">loc_GnssData</a>. See <a href="#">loc_GnssData</a> for more information</li> </ul>
<i>pCellDb</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">loc_CellDb</a>. See <a href="#">loc_CellDb</a> for more information</li> </ul>
<i>pClkInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">loc_ClkInfo</a>. See <a href="#">loc_ClkInfo</a> for more information</li> </ul>
<i>pBdsSVInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to struct <a href="#">loc_BdsSVInfo</a>. See <a href="#">loc_BdsSVInfo</a> for more information</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack delete assist data request result.</li> </ul>

## 8.492.2 Field Documentation

8.492.2.1 `loc_BdsSVInfo*` `pack_loc_Delete_Assist_Data_t::pBdsSVInfo`8.492.2.2 `loc_CellDb*` `pack_loc_Delete_Assist_Data_t::pCellDb`8.492.2.3 `loc_ClkInfo*` `pack_loc_Delete_Assist_Data_t::pClkInfo`8.492.2.4 `loc_GnssData*` `pack_loc_Delete_Assist_Data_t::pGnssData`8.492.2.5 `loc_SVInfo*` `pack_loc_Delete_Assist_Data_t::pSVInfo`8.492.2.6 `uint16_t` `pack_loc_Delete_Assist_Data_t::Tlvresult`

## 8.493 pack\_loc\_EventRegister\_t Struct Reference

## Data Fields

- `uint64_t` [eventRegister](#)
- `uint16_t` [Tlvresult](#)

## 8.493.1 Detailed Description

This structure contains the Parameter for RegisterEvents

## Parameters

<i>eventRegister</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. These reports are sent at a 1 Hz rate.</li> <li>– 0x00000004 - to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.</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 engine state report event indications.</li> <li>– 0x00000100 - 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 (accelerometer, gyroscope, etc.).</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. Reports are generated only for the GNSS satellite constellations that are enabled using QMI_LOC_SET_GNSS_CONSTELL_REPORT_CONFIG.</li> </ul> </li> </ul>
----------------------	--

## Note

Multiple events can be registered by OR the individual masks and sending them in this TLV. All unused bits in this mask must be set to 0.

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack result.</li> </ul>
------------------	--

## 8.493.2 Field Documentation

8.493.2.1 uint64\_t pack\_loc\_EventRegister\_t::eventRegister

8.493.2.2 uint16\_t pack\_loc\_EventRegister\_t::Tlvresult

## 8.494 pack\_loc\_SetExtPowerState\_t Struct Reference

## Data Fields

- uint32\_t [extPowerState](#)
- uint16\_t [Tlvresult](#)

## 8.494.1 Detailed Description

This structure contains the Parameter External Power Source State pack.

## Parameters

<i>extPowerState</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>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack result.</li> </ul>

## 8.494.2 Field Documentation

8.494.2.1 uint32\_t pack\_loc\_SetExtPowerState\_t::extPowerState

8.494.2.2 uint16\_t pack\_loc\_SetExtPowerState\_t::Tlvresult

## 8.495 pack\_loc\_SetOperationMode\_t Struct Reference

## Data Fields

- uint32\_t [mode](#)
- uint16\_t [Tlvresult](#)

### 8.495.1 Detailed Description

This structure contains Set Operation Mode pack

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>0 - Default Mode.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack result.</li> </ul>

### 8.495.2 Field Documentation

8.495.2.1 uint32\_t pack\_loc\_SetOperationMode\_t::mode

8.495.2.2 uint16\_t pack\_loc\_SetOperationMode\_t::Tlvresult

## 8.496 pack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

### Data Fields

- uint32\_t [xid](#)
- uint16\_t [Tlvresult](#)

### 8.496.1 Detailed Description

This structure contains Set Operation Mode pack

#### Parameters

<i>xid</i>	<ul style="list-style-type: none"> <li>Identifies the transaction.</li> <li>The transaction ID is returned in the Get Best Available Position indication.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack result.</li> </ul>

### 8.496.2 Field Documentation

8.496.2.1 uint16\_t pack\_loc\_SLQSLOCGetBestAvailPos\_t::Tlvresult

8.496.2.2 uint32\_t pack\_loc\_SLQSLOCGetBestAvailPos\_t::xid

## 8.497 pack\_loc\_Start\_t Struct Reference

### Data Fields

- uint8\_t [SessionId](#)
- uint32\_t \* [pRecurrenceType](#)
- uint32\_t \* [pHorizontalAccuracyLvl](#)

- uint32\_t \* [pIntermediateReportState](#)
- uint32\_t \* [pMinIntervalTime](#)
- [loc\\_LocApplicationInfo](#) \* [pApplicationInfo](#)
- uint32\_t \* [pConfigAltitudeAssumed](#)
- uint16\_t [Tlvresult](#)

### 8.497.1 Detailed Description

This structure contains the LOC Start pack

#### 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="#">loc_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>

## 8.497.2 Field Documentation

8.497.2.1 `loc_LocApplicationInfo*` `pack_loc_Start_t::pApplicationInfo`

8.497.2.2 `uint32_t*` `pack_loc_Start_t::pConfigAltitudeAssumed`

8.497.2.3 `uint32_t*` `pack_loc_Start_t::pHorizontalAccuracyLvl`

8.497.2.4 `uint32_t*` `pack_loc_Start_t::pIntermediateReportState`

8.497.2.5 `uint32_t*` `pack_loc_Start_t::pMinIntervalTime`

8.497.2.6 `uint32_t*` `pack_loc_Start_t::pRecurrenceType`

8.497.2.7 `uint8_t` `pack_loc_Start_t::SessionId`

8.497.2.8 `uint16_t` `pack_loc_Start_t::Tlvresult`

## 8.498 `pack_loc_Stop_t` Struct Reference

### Data Fields

- `uint8_t` [SessionId](#)
- `uint16_t` [Tlvresult](#)

### 8.498.1 Detailed Description

This structure contains Stop LOC pack

#### 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>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> </ul>

## 8.498.2 Field Documentation

8.498.2.1 `uint8_t` `pack_loc_Stop_t::SessionId`

8.498.2.2 `uint16_t` `pack_loc_Stop_t::Tlvresult`

## 8.499 `pack_nas_SetACCOLC_t` Struct Reference

### Data Fields

- `int8_t` [spc](#) [6]
- `uint8_t` [accolc](#)

### 8.499.1 Detailed Description

#### Parameters

<i>spc</i>	servcie programming code
<i>accolc</i>	accolc

### 8.499.2 Field Documentation

8.499.2.1 `uint8_t pack_nas_SetACCOLC_t::accolc`

8.499.2.2 `int8_t pack_nas_SetACCOLC_t::spc[6]`

## 8.500 pack\_nas\_SetNetworkPreference\_t Struct Reference

### Data Fields

- `uint32_t TechnologyPref`
- `uint32_t Duration`
- `uint16_t Tlvresult`

### 8.500.1 Detailed Description

#### Parameters

<i>TechnologyPref[IN]</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>Duration[IN]</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>Tlvresult</i>	<ul style="list-style-type: none"> <li>• pack result</li> </ul>

### 8.500.2 Field Documentation

8.500.2.1 uint32\_t pack\_nas\_SetNetworkPreference\_t::Duration

8.500.2.2 uint32\_t pack\_nas\_SetNetworkPreference\_t::TechnologyPref

8.500.2.3 uint16\_t pack\_nas\_SetNetworkPreference\_t::Tlvresult

## 8.501 pack\_nas\_SLQSGetPLMNName\_t Struct Reference

### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- uint8\_t \* [pMncPcsStatus](#)

### 8.501.1 Detailed Description

#### 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>
<i>pMncPcsStatus</i>	<ul style="list-style-type: none"> <li>• MNC PCS Digit Include Status</li> <li>• Used to interpret the length of the corresponding MNC reported in the PLMN TLV(0x01).</li> <li>• Values <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> </li> </ul>

#### Note

If pMncPcsStatus is not present, an MNC smaller than 100 is assumed to be a two-digit value, and an MNC greater than or equal to 100 is assumed to be a three digit value.

### 8.501.2 Field Documentation

8.501.2.1 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mcc

8.501.2.2 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mnc

8.501.2.3 uint8\_t\* pack\_nas\_SLQSGetPLMNName\_t::pMncPcsStatus

## 8.502 pack\_nas\_SLQSInitiateNetworkRegistration\_t Struct Reference

### Data Fields

- uint32\_t [regAction](#)
- [nas\\_MNRInfo](#) \* [pMNRInfo](#)

- uint32\_t \* [pChangeDuration](#)
- uint8\_t \* [pMncPcsDigitStatus](#)

### 8.502.1 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>– AUTO_REGISTER - Device registers according to its provisioning and optional parameters supplied with the command are ignored.</li> <li>– MANUAL_REGISTER - Device registers to a specified network and the optional Manual Network Register Information parameter pMNRInfo must also be included for the command to process successfully and supported only for 3GPP.</li> </ul> </li> </ul>
<i>pMNRInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• Pointer to structure <a href="#">MNRInfo</a> <ul style="list-style-type: none"> <li>– See <a href="#">nas_MNRInfo</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>– 0x00 - Power cycle - Remains active until the next device power cycle</li> <li>– 0x01 - 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>

### 8.502.2 Field Documentation

8.502.2.1 uint32\_t\* pack\_nas\_SLQSInitiateNetworkRegistration\_t::pChangeDuration

8.502.2.2 uint8\_t\* pack\_nas\_SLQSInitiateNetworkRegistration\_t::pMncPcsDigitStatus

8.502.2.3 nas\_MNRInfo\* pack\_nas\_SLQSInitiateNetworkRegistration\_t::pMNRInfo

8.502.2.4 uint32\_t pack\_nas\_SLQSInitiateNetworkRegistration\_t::regAction

## 8.503 pack\_nas\_SLQSNasConfigSigInfo2\_t Struct Reference

### Data Fields

- [nas\\_CDMARSSIThresh](#) \* [pCDMARSSIThresh](#)
- uint16\_t \* [pCDMARSSIDelta](#)
- [nas\\_CDMAECIOThresh](#) \* [pCDMAECIOThresh](#)
- uint16\_t \* [pCDMAECIODelta](#)
- [nas\\_HDRRSSIThresh](#) \* [pHDRRSSIThresh](#)
- uint16\_t \* [pHDRRSSIDelta](#)
- [nas\\_HDRECIOThresh](#) \* [pHDRECIOThresh](#)

- uint16\_t \* pHDRECIODelta
- nas\_HDRSINRThreshold \* pHDRSINRThresh
- uint16\_t \* pHDRSINRDelta
- nas\_HDRIOTresh \* pHDRIOTresh
- uint16\_t \* pHDRIODelta
- nas\_GSMRSSIthresh \* pGSMRSSIthresh
- uint16\_t \* pGSMRSSIDelta
- nas\_WCDMARSSIthresh \* pWCDMARSSIthresh
- uint16\_t \* pWCDMARSSIDelta
- nas\_WCDMAECIOthresh \* pWCDMAECIOthresh
- uint16\_t \* pWCDMAECIODelta
- nas\_LTERSSIthresh \* pLTERSSIthresh
- uint16\_t \* pLTERSSIDelta
- nas\_LTESNRThreshold \* pLTESNRThresh
- uint16\_t \* pLTESNRDelta
- nas\_LTERSRQThresh \* pLTERSRQThresh
- uint16\_t \* pLTERSRQDelta
- nas\_LTERSRPThresh \* pLTERSRPThresh
- uint16\_t \* pLTERSRPDelta
- nas\_LTESigRptConfig \* pLTESigRptConfig
- nas\_TDSCDMARSCPTthresh \* pTDSCDMARSCPTthresh
- uint16\_t \* pTDSCDMARSCPDelta
- nas\_TDSCDMARSSIthresh \* pTDSCDMARSSIthresh
- float \* pTDSCDMARSSIDelta
- nas\_TDSCDMAECIOthresh \* pTDSCDMAECIOthresh
- float \* pTDSCDMAECIODelta
- nas\_TDSCDMASINRThresh \* pTDSCDMASINRThresh
- float \* pTDSCDMASINRDelta

### 8.503.1 Detailed Description

#### Parameters

<i>pCDMARSSI- Thresh</i>	<ul style="list-style-type: none"> <li>• CDMA RSSI threshold List</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> </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> </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>pHdRECIO- Thresh</i>	<ul style="list-style-type: none"> <li>HDR ECIO Threshold List</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> </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>pHdRIOTresh</i>	<ul style="list-style-type: none"> <li>HDR IO Threshold List</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> </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> </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>pLTERSNR- Thresh</i>	<ul style="list-style-type: none"> <li>• LTE SNR Threshold List</li> </ul>
<i>pLTERSNRDelta</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> </ul>
<i>pLTERSRQ- 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> </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> </ul>
<i>pTDSCDMARS- CPThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA RSCP Threshold List</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> </ul>
<i>pTDSCDMARS- SIDelta</i>	<ul style="list-style-type: none"> <li>• RSSI delta (in dBm) used by TD-SCDMA.</li> </ul>
<i>pTDSCDMAECI- OThresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA ECIO Threshold List</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> </ul>
<i>pTDSCDMASIN- RDelta</i>	<ul style="list-style-type: none"> <li>• SINR delta (in dB) used by TD-SCDMA.</li> </ul>

### 8.503.2 Field Documentation

- 8.503.2.1 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMAECIODelta
- 8.503.2.2 nas\_CDMAECIOThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMAECIOThresh
- 8.503.2.3 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMARSSIDelta
- 8.503.2.4 nas\_CDMARSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMARSSIThresh
- 8.503.2.5 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pGSMRSSIDelta
- 8.503.2.6 nas\_GSMRSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pGSMRSSIThresh
- 8.503.2.7 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHDRECIODelta
- 8.503.2.8 nas\_HDRECIOThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHDRECIOThresh
- 8.503.2.9 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHDRIODelta
- 8.503.2.10 nas\_HDRIOThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHDRIOThresh
- 8.503.2.11 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHRRSSIDelta
- 8.503.2.12 nas\_HDRRSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHRRSSIThresh
- 8.503.2.13 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHRSINRDelta
- 8.503.2.14 nas\_HDRSINRThreshold\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHRSINRThresh
- 8.503.2.15 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSRPDelta
- 8.503.2.16 nas\_LTERSRPThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSRPThresh
- 8.503.2.17 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSRQDelta
- 8.503.2.18 nas\_LTERSRQThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSRQThresh
- 8.503.2.19 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSSIDelta
- 8.503.2.20 nas\_LTERSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTERSSIThresh
- 8.503.2.21 nas\_LTESigRptConfig\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTESigRptConfig
- 8.503.2.22 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTESNRDelta
- 8.503.2.23 nas\_LTESNRThreshold\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pLTESNRThresh
- 8.503.2.24 float\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pTDSCDMAECIODelta
- 8.503.2.25 nas\_TDSCDMAECIOThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pTDSCDMAECIOThresh
- 8.503.2.26 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pTDSCDMARSCPDelta
- 8.503.2.27 nas\_TDSCDMARSCPThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pTDSCDMARSCPThresh
- 8.503.2.28 float\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pTDSCDMARSSIDelta

8.503.2.29 `nas_TDSCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIThresh`

8.503.2.30 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta`

8.503.2.31 `nas_TDSCDMASINRThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`

8.503.2.32 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIODelta`

8.503.2.33 `nas_WCDMAECIOThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMAECIOThresh`

8.503.2.34 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`

8.503.2.35 `nas_WCDMARSSIThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIThresh`

## 8.504 `pack_nas_SLQSNasIndicationRegisterExt_t` Struct Reference

### Data Fields

- `uint8_t *` [pSystemSelectionInd](#)
- `uint8_t *` [pDDTMInd](#)
- `uint8_t *` [pServingSystemInd](#)
- `uint8_t *` [pDualStandByPrefInd](#)
- `uint8_t *` [pSubscriptionInfoInd](#)
- `uint8_t *` [pNetworkTimeInd](#)
- `uint8_t *` [pSysInfoInd](#)
- `uint8_t *` [pSignalStrengthInd](#)
- `uint8_t *` [pErrorRateInd](#)
- `uint8_t *` [pHDRNewUATIAssInd](#)
- `uint8_t *` [pHDRSessionCloseInd](#)
- `uint8_t *` [pManagedRoamingInd](#)
- `uint8_t *` [pLTECphyCa](#)

### 8.504.1 Detailed Description

#### Parameters

<i>pSystem-SelectionInd</i>	[Optional] <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="#">tFNDDataCapabilities</a> and <a href="#">tFNServingSystem</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pDDTMInd</i>	[Optional] <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.  <a href="#">tFNDDTM</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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.  <a href="#">tFNBandPreference</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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.  <a href="#">tFNDualStandByPref</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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.  <a href="#">tFNSubscriptionInfo</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pNetworkTime-Ind</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.  <a href="#">tFNNetworkTime</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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.  <a href="#">tFNSysInfo</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSignalStrength-Ind</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.  <a href="#">tFNSigInfo</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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.  <a href="#">tFNErrRate</a> <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pHDRNewUATI-AssInd</i>	[Optional] <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. tFNHDRUATIUpdate             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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. tFNHDRSessionClose             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </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. tFNManagedRoaming             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pLTECphyCa</i>	[Optional] <ul style="list-style-type: none"> <li>LTE Physical Carrier Aggregation Information. The following callbacks would not be invoked if the indication is disabled. tFNManagedRoaming             <ul style="list-style-type: none"> <li>– 0x00 - Disable (default value)</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

## 8.504.2 Field Documentation

8.504.2.1 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pDDTMInd

8.504.2.2 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pDualStandByPrefInd

8.504.2.3 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pErrorRateInd

8.504.2.4 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pHDRNewUATIAssInd

8.504.2.5 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pHDRSessionCloseInd

8.504.2.6 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pLTECphyCa

8.504.2.7 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pManagedRoamingInd

8.504.2.8 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pNetworkTimeInd

8.504.2.9 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pServingSystemInd

8.504.2.10 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSignalStrengthInd

8.504.2.11 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSubscriptionInfoInd

8.504.2.12 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSysInfoInd

8.504.2.13 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterExt\_t::pSystemSelectionInd

## 8.505 pack\_nas\_SLQSNasSwiOTAMessageCallback\_t Struct Reference

### Data Fields

- uint8\_t [lteEsmUI](#)
- uint8\_t [lteEsmDI](#)
- uint8\_t [lteEmmUI](#)
- uint8\_t [lteEmmDI](#)
- uint8\_t [gsmUmtsUI](#)
- uint8\_t [gsmUmtsDI](#)
- uint8\_t \* [pRankIndicatorInd](#)

### 8.505.1 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>
<i>pRankIndicatorInd</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report Rank Indicator messages</li> </ul>

### 8.505.2 Field Documentation

- 8.505.2.1 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsDI`
- 8.505.2.2 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::gsmUmtsUI`
- 8.505.2.3 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmDI`
- 8.505.2.4 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEmmUI`
- 8.505.2.5 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmDI`
- 8.505.2.6 `uint8_t pack_nas_SLQSNasSwiOTAMessageCallback_t::lteEsmUI`
- 8.505.2.7 `uint8_t* pack_nas_SLQSNasSwiOTAMessageCallback_t::pRankIndicatorInd`

## 8.506 `pack_nas_SLQSSetSignalStrengthsCallback_t` Struct Reference

### Data Fields

- `uint8_t bEnable`
- `nas_SLQSSignalStrengthsIndReq * pSigIndReq`

### 8.506.1 Detailed Description

#### Parameters

<i>bEnable</i>	0/1 to disable/enable RSSI signal strength indication
<i>pSigIndReq</i>	parameters to control signal strength indication

### 8.506.2 Field Documentation

- 8.506.2.1 `uint8_t pack_nas_SLQSSetSignalStrengthsCallback_t::bEnable`
- 8.506.2.2 `nas_SLQSSignalStrengthsIndReq* pack_nas_SLQSSetSignalStrengthsCallback_t::pSigIndReq`

## 8.507 `pack_nas_SLQSSetSysSelectionPref_t` Struct Reference

### Data Fields

- `uint8_t * pEmerMode`
- `uint16_t * pModePref`
- `uint64_t * pBandPref`
- `uint16_t * pPRLPref`
- `uint16_t * pRoamPref`
- `uint64_t * pLTEBandPref`
- `struct nas_netSelectionPref * pNetSelPref`
- `uint8_t * pChgDuration`
- `uint8_t * pMNCIncPCSDigStat`
- `uint32_t * pSrvDomainPref`
- `uint32_t * pGWAcqOrderPref`
- `uint64_t * pTdsdmaBandPref`
- `struct nas_acqOrderPref * pAcqOrderPref`
- `uint32_t * pSrvRegRestriction`
- `struct nas_CSGID * pCSGID`
- `unsigned char * pRAT`

### 8.507.1 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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</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> <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> <li>– All other bits are reserved</li> </ul> </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>

## 8.507.2 Field Documentation

8.507.2.1 struct nas\_acqOrderPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pAcqOrderPref

8.507.2.2 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pBandPref

8.507.2.3 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pChgDuration

8.507.2.4 struct nas\_CSGID\* pack\_nas\_SLQSSetSysSelectionPref\_t::pCSGID

8.507.2.5 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pEmerMode

8.507.2.6 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pGWAcqOrderPref

- 8.507.2.7 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pLTEBandPref
- 8.507.2.8 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pMNCIncPCSDigStat
- 8.507.2.9 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pModePref
- 8.507.2.10 struct nas\_netSelectionPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pNetSelPref
- 8.507.2.11 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pPRLPref
- 8.507.2.12 unsigned char\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRAT
- 8.507.2.13 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRoamPref
- 8.507.2.14 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvDomainPref
- 8.507.2.15 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvRegRestriction
- 8.507.2.16 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pTdsdmaBandPref

## 8.508 pack\_qmi\_t Struct Reference

### Data Fields

- uint16\_t [xid](#)
- int [timeout](#)
- uint16\_t [msgid](#)
- uint8\_t [svc](#)

### 8.508.1 Detailed Description

qmi request context

#### Parameters

in	<i>xid</i>	transaction id
out	<i>timeout</i>	timeout recommended in seconds
out	<i>msgid</i>	message id
out	<i>svc</i>	qmi service

### 8.508.2 Field Documentation

- 8.508.2.1 uint16\_t pack\_qmi\_t::msgid
- 8.508.2.2 uint8\_t pack\_qmi\_t::svc
- 8.508.2.3 int pack\_qmi\_t::timeout
- 8.508.2.4 uint16\_t pack\_qmi\_t::xid

## 8.509 pack\_qos\_SLQSQosSwiReadApnExtraParams\_t Struct Reference

## Data Fields

- uint32\_t [apnId](#)

### 8.509.1 Detailed Description

Structure that contains the APN ID to obtain extra APN parameters

#### Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none"> <li>• APN id</li> </ul>
-------------------	--

### 8.509.2 Field Documentation

8.509.2.1 uint32\_t pack\_qos\_SLQSQosSwiReadApnExtraParams\_t::apnId

## 8.510 pack\_qos\_SLQSQosSwiReadDataStats\_t Struct Reference

## Data Fields

- uint32\_t [apnId](#)

### 8.510.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

#### Parameters

<i>apnId</i> [IN]	<ul style="list-style-type: none"> <li>• APN id</li> </ul>
-------------------	--

### 8.510.2 Field Documentation

8.510.2.1 uint32\_t pack\_qos\_SLQSQosSwiReadDataStats\_t::apnId

## 8.511 pack\_qos\_SLQSSetQosEventCallback\_t Struct Reference

## Data Fields

- uint8\_t [enable](#)

### 8.511.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

## Parameters

<i>enable</i> [IN]	<ul style="list-style-type: none"> <li>• 1 - Enable QoS event reporting</li> <li>• 0 - Disable QoS event reporting</li> </ul>
--------------------	---

## 8.511.2 Field Documentation

8.511.2.1 uint8\_t pack\_qos\_SLQSSetQosEventCallback\_t::enable

## 8.512 pack\_sms\_SendSMS\_t Struct Reference

## Data Fields

- uint32\_t [messageFormat](#)
- uint32\_t [messageSize](#)
- uint8\_t \* [pMessage](#)
- uint8\_t \* [pLinktimer](#)

## 8.512.1 Detailed Description

## 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>pLinktimer</i>	<ul style="list-style-type: none"> <li>• GW SMS link open for the specified number of second</li> </ul>
<i>pMessage</i>	<ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>

## 8.512.2 Field Documentation

8.512.2.1 uint32\_t pack\_sms\_SendSMS\_t::messageFormat

8.512.2.2 uint32\_t pack\_sms\_SendSMS\_t::messageSize

8.512.2.3 uint8\_t\* pack\_sms\_SendSMS\_t::pLinktimer

8.512.2.4 uint8\_t\* pack\_sms\_SendSMS\_t::pMessage

## 8.513 pack\_sms\_SetNewSMSCallback\_t Struct Reference

## Data Fields

- enum [eqmiCbkJSetStatus status](#)

### 8.513.1 Detailed Description

#### Parameters

<i>status</i>	callback parameter
---------------	--------------------

### 8.513.2 Field Documentation

8.513.2.1 enum [eqmiCbkJSetStatus pack\\_sms\\_SetNewSMSCallback\\_t::status](#)

## 8.514 pack\_sms\_SLQSDelateSMS\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pMessageIndex](#)
- uint32\_t \* [pMessageTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.514.1 Detailed Description

#### 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>	<ul style="list-style-type: none"> <li>• (Optional) message index</li> </ul>
<i>pMessageTag</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>pMessageMode</i>	<ul style="list-style-type: none"> <li>• (Optional) message mode</li> <li>• this must be included if the device is capable of supporting more than one protocol</li> <li>• e.g. CDMA and GW <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>

### 8.514.2 Field Documentation

8.514.2.1 uint32\_t\* pack\_sms\_SLQSDelSDeleteSMS\_t::pMessageIndex

8.514.2.2 uint8\_t\* pack\_sms\_SLQSDelSDeleteSMS\_t::pMessageMode

8.514.2.3 uint32\_t\* pack\_sms\_SLQSDelSDeleteSMS\_t::pMessageTag

8.514.2.4 uint32\_t pack\_sms\_SLQSDelSDeleteSMS\_t::storageType

## 8.515 pack\_sms\_SLQSGetSMS\_t Struct Reference

### Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)
- uint8\_t \* [pMessageMode](#)

### 8.515.1 Detailed Description

#### 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>pMessageMode</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>

### 8.515.2 Field Documentation

8.515.2.1 uint32\_t pack\_sms\_SLQSGetSMS\_t::messageIndex

8.515.2.2 uint8\_t\* pack\_sms\_SLQSGetSMS\_t::pMessageMode

8.515.2.3 uint32\_t pack\_sms\_SLQSGetSMS\_t::storageType

## 8.516 pack\_sms\_SLQSGetSMSList\_t Struct Reference

### Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pRequestedTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.516.1 Detailed Description

## 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>requestedTag</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>messageMode</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>

## 8.516.2 Field Documentation

8.516.2.1 uint8\_t\* pack\_sms\_SLQSGetSMSList\_t::pMessageMode

8.516.2.2 uint32\_t\* pack\_sms\_SLQSGetSMSList\_t::pRequestedTag

8.516.2.3 uint32\_t pack\_sms\_SLQSGetSMSList\_t::storageType

## 8.517 pack\_sms\_SLQSMModifySMSStatus\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)
- uint32\_t [messageTag](#)
- uint8\_t \* [pMessageMode](#)

## 8.517.1 Detailed Description

## 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>pMessageMode</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>
---------------------	--

## 8.517.2 Field Documentation

8.517.2.1 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageIndex

8.517.2.2 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageTag

8.517.2.3 uint8\_t\* pack\_sms\_SLQSMModifySMSStatus\_t::pMessageMode

8.517.2.4 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::storageType

## 8.518 pack\_swiloc\_SwiLocSetAutoStart\_t Struct Reference

### Data Fields

- uint8\_t [function](#)
- int [set\\_function](#)
- uint8\_t [fix\\_type](#)
- int [set\\_fix\\_type](#)
- uint8\_t [max\\_time](#)
- int [set\\_max\\_time](#)
- uint32\_t [max\\_dist](#)
- int [set\\_max\\_dist](#)
- uint32\_t [fix\\_rate](#)
- int [set\\_fix\\_rate](#)

### 8.518.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

#### Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>• Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> <li>– 0 - disabled</li> <li>– 1 - At bootup</li> <li>– 2 - When NMEA port is opened</li> </ul> </li> </ul>
<i>set_function</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>• Type of GNSS fix: <ul style="list-style-type: none"> <li>– 1 - Default Engine mode</li> <li>– 2 - MS-Based</li> <li>– 3 - MS-Assisted</li> <li>– 4 - Standalone</li> </ul> </li> </ul>

<i>set_fix_type</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>• Maximum time allowed for the receiver to get a fix in seconds</li> <li>• Valid range: 1-255</li> </ul>
<i>set_max_time</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>max_dist</i>	<ul style="list-style-type: none"> <li>• Maximum uncertainty of a fix measured by distance in meters</li> <li>• Valid range: 1 - 4294967280</li> </ul>
<i>set_max_dist</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>• Time between fixes in seconds</li> <li>• Valid range: 1–65535</li> </ul>
<i>set_fix_rate</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>

## 8.518.2 Field Documentation

8.518.2.1 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::fix_rate`

8.518.2.2 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::fix_type`

8.518.2.3 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::function`

8.518.2.4 `uint32_t pack_swiloc_SwiLocSetAutoStart_t::max_dist`

8.518.2.5 `uint8_t pack_swiloc_SwiLocSetAutoStart_t::max_time`

8.518.2.6 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_rate`

8.518.2.7 `int pack_swiloc_SwiLocSetAutoStart_t::set_fix_type`

8.518.2.8 `int pack_swiloc_SwiLocSetAutoStart_t::set_function`

8.518.2.9 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_dist`

8.518.2.10 `int pack_swiloc_SwiLocSetAutoStart_t::set_max_time`

## 8.519 `pack_swioama_SLQSOMADMCancelSession_t` Struct Reference

## Data Fields

- uint32\_t [sessionType](#)

### 8.519.1 Detailed Description

Structure that contains the session type for OMA cancel session command

#### 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>– 0xFF - Cancel any active OMADM session</li></ul></li></ul>
------------------------	--

### 8.519.2 Field Documentation

8.519.2.1 uint32\_t pack\_swisma\_SLQSOMADMCancelSession\_t::sessionType

## 8.520 pack\_swisma\_SLQSOMADMGetSessionInfo\_t Struct Reference

## Data Fields

- uint32\_t [SessionType](#)

### 8.520.1 Detailed Description

Structure that contains the session type for OMA get session info command

#### Parameters

<i>SessionType[IN]</i>	<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 no active sessions are available, then previous OMADM session info is returned</li></ul></li></ul>
------------------------	--

### 8.520.2 Field Documentation

8.520.2.1 uint32\_t pack\_swisma\_SLQSOMADMGetSessionInfo\_t::SessionType

## 8.521 pack\_swisma\_SLQSOMADMSendSelection\_t Struct Reference

## Data Fields

- uint32\_t [selection](#)
- uint32\_t \* [pDeferTime](#)
- uint32\_t \* [pRejectReason](#)

### 8.521.1 Detailed Description

Structure containing the OMA DM selection

#### 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>

### 8.521.2 Field Documentation

8.521.2.1 uint32\_t\* pack\_swioama\_SLQSOMADMSendSelection\_t::pDeferTime

8.521.2.2 uint32\_t\* pack\_swioama\_SLQSOMADMSendSelection\_t::pRejectReason

8.521.2.3 uint32\_t pack\_swioama\_SLQSOMADMSendSelection\_t::selection

## 8.522 pack\_swioama\_SLQSOMADMSetSettings\_t Struct Reference

#### Data Fields

- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t \* [pAutosdm](#)
- uint8\_t \* [pFwAutoCheck](#)

### 8.522.1 Detailed Description

Structure containing the OMA DM settings to be set on the device This maps to structure SLQSOMADMSettings-ReqParams3

#### 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 autodownload FALSE</li> <li>0x01 - Firmware autodownload 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 autoupdate FALSE</li> <li>– 0x01 - Firmware autoupdate 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>

## 8.522.2 Field Documentation

8.522.2.1 uint8\_t pack\_swioama\_SLQSOMADMSetSettings\_t::FOTAdownload

8.522.2.2 uint8\_t pack\_swioama\_SLQSOMADMSetSettings\_t::FOTAUpdate

8.522.2.3 uint8\_t\* pack\_swioama\_SLQSOMADMSetSettings\_t::pAutosdm

8.522.2.4 uint8\_t\* pack\_swioama\_SLQSOMADMSetSettings\_t::pFwAutoCheck

## 8.523 pack\_swioama\_SLQSOMADMStartSession\_t Struct Reference

### Data Fields

- uint32\_t [sessionType](#)

### 8.523.1 Detailed Description

Structure that contains the session type for OMA start session command

#### 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>
------------------------	---

## 8.523.2 Field Documentation

8.523.2.1 uint32\_t pack\_swioama\_SLQSOMADMStartSession\_t::sessionType

## 8.524 pack\_uim\_ChangePin\_t Struct Reference

### Data Fields

- [uim\\_encryptedPIN1](#) EncryptedPIN1
- [uint32\\_t \\* pIndicationToken](#)
- [uint8\\_t \\* pKeyReferenceID](#)
- [uim\\_sessionInformation](#) sessionInfo
- [uim\\_changeUIMPIN](#) changePIN
- [uint16\\_t Tlvresult](#)

### 8.524.1 Detailed Description

This structure contains information of the request parameters associated with a Change 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="#">changePIN</a>	<ul style="list-style-type: none"> <li>• See <a href="#">changeUIMPIN</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> <li>• Valid Values               <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.524.2 Field Documentation

8.524.2.1 [uim\\_changeUIMPIN](#) [pack\\_uim\\_ChangePin\\_t::changePIN](#)

8.524.2.2 [uim\\_encryptedPIN1](#) [pack\\_uim\\_ChangePin\\_t::EncryptedPIN1](#)

8.524.2.3 [uint32\\_t\\*](#) [pack\\_uim\\_ChangePin\\_t::pIndicationToken](#)

8.524.2.4 [uint8\\_t\\*](#) [pack\\_uim\\_ChangePin\\_t::pKeyReferenceID](#)

8.524.2.5 [uim\\_sessionInformation](#) [pack\\_uim\\_ChangePin\\_t::sessionInfo](#)

8.524.2.6 uint16\_t pack\_uim\_ChangePin\_t::Tlvresult

## 8.525 pack\_uim\_ReadTransparent\_t Struct Reference

### Data Fields

- [uim\\_sessionInformation](#) sessionInfo
- [uim\\_fileInfo](#) fileIndex
- [uim\\_readTransparentInfo](#) readTransparent
- uint32\_t \* [pIndicationToken](#)
- uint8\_t \* [pEncryptData](#)
- uint16\_t [Tlvresult](#)

### 8.525.1 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>
<a href="#">fileIndex</a>	<ul style="list-style-type: none"> <li>• See <a href="#">fileInfo</a> for more information.</li> </ul>
<a href="#">readTransparent</a>	<ul style="list-style-type: none"> <li>• See <a href="#">readTransparentInfo</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> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>
<a href="#">pEncrypt-Data(optional)</a>	<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.

### 8.525.2 Field Documentation

8.525.2.1 uim\_fileInfo pack\_uim\_ReadTransparent\_t::fileIndex

8.525.2.2 uint8\_t\* pack\_uim\_ReadTransparent\_t::pEncryptData

8.525.2.3 uint32\_t\* pack\_uim\_ReadTransparent\_t::pIndicationToken

8.525.2.4 `uim_readTransparentInfo` `pack_uim_ReadTransparent_t::readTransparent`

8.525.2.5 `uim_sessionInformation` `pack_uim_ReadTransparent_t::sessionInfo`

8.525.2.6 `uint16_t` `pack_uim_ReadTransparent_t::Tlvresult`

## 8.526 `pack_uim_SetPinProtection_t` Struct Reference

### Data Fields

- `uim_encryptedPIN1` `EncryptedPIN1`
- `uint32_t * pIndicationToken`
- `uint8_t * pKeyReferenceID`
- `uim_sessionInformation` `sessionInfo`
- `uim_setPINProtection` `pinProtection`
- `uint16_t` `Tlvresult`

### 8.526.1 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="#">uim_sessionInformation</a> for more information.</li> </ul>
<i><a href="#">pinProtection</a></i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_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="#">pIndicationToken(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> <li>• Valid Values               <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.526.2 Field Documentation

8.526.2.1 `uim_encryptedPIN1` `pack_uim_SetPinProtection_t::EncryptedPIN1`

8.526.2.2 `uint32_t*` `pack_uim_SetPinProtection_t::pIndicationToken`

8.526.2.3 uim\_setPINProtection pack\_uim\_SetPinProtection\_t::pinProtection

8.526.2.4 uint8\_t\* pack\_uim\_SetPinProtection\_t::pKeyReferenceID

8.526.2.5 uim\_sessionInformation pack\_uim\_SetPinProtection\_t::sessionInfo

8.526.2.6 uint16\_t pack\_uim\_SetPinProtection\_t::Tlvresult

## 8.527 pack\_uim\_SLQSUIEventRegister\_t Struct Reference

### Data Fields

- uint32\_t [eventMask](#)

### 8.527.1 Detailed Description

#### Parameters

<i>eventMask</i>	<ul style="list-style-type: none"><li>- bit 1 - card status</li><li>• bit 4 - physical slot status</li></ul>
------------------	--

### 8.527.2 Field Documentation

8.527.2.1 uint32\_t pack\_uim\_SLQSUIEventRegister\_t::eventMask

## 8.528 pack\_uim\_SLQSUIMSwitchSlot\_t Struct Reference

### Data Fields

- uint8\_t [bLogicalSlot](#)
- uint32\_t [ulPhysicalSlot](#)

### 8.528.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

#### Parameters

<i>bLogicalSlot</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><li>– 3 - Slot 3</li><li>– 4 - Slot 4</li><li>– 5 - Slot 5</li></ul></li></ul>
---------------------	--

<i>ulPhysicalSlot</i>	<ul style="list-style-type: none"> <li>• 1 - Slot 1</li> <li>• 2 - Slot 2</li> <li>• 3 - Slot 3</li> <li>• 4 - Slot 4</li> <li>• 5 - Slot 5</li> </ul>
-----------------------	--

## 8.528.2 Field Documentation

8.528.2.1 `uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot`

8.528.2.2 `uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot`

## 8.529 `pack_uim_UnblockPin_t` Struct Reference

### Data Fields

- [uim\\_encryptedPIN1](#) EncryptedPIN1
- `uint32_t *` [pIndicationToken](#)
- `uint8_t *` [pKeyReferenceID](#)
- [uim\\_sessionInformation](#) sessionInfo
- [uim\\_unblockUIMPIN](#) pinProtection
- `uint16_t` [Tlvresult](#)

### 8.529.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN API.

#### Parameters

<i>EncryptedPIN1</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> </ul>
<i><a href="#">sessionInfo</a></i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>pinProtection</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_unblockUIMPIN</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> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Result of operation in response. Indication will not be generated by the modem</li> <li>– Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>
-----------------------------------	---

## 8.529.2 Field Documentation

8.529.2.1 uim\_encryptedPIN1 pack\_uim\_UnblockPin\_t::EncryptedPIN1

8.529.2.2 uint32\_t\* pack\_uim\_UnblockPin\_t::pIndicationToken

8.529.2.3 uim\_unblockUIMPIN pack\_uim\_UnblockPin\_t::pinProtection

8.529.2.4 uint8\_t\* pack\_uim\_UnblockPin\_t::pKeyReferenceID

8.529.2.5 uim\_sessionInformation pack\_uim\_UnblockPin\_t::sessionInfo

8.529.2.6 uint16\_t pack\_uim\_UnblockPin\_t::Tlvresult

## 8.530 pack\_uim\_VerifyPin\_t Struct Reference

### Data Fields

- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint8\\_t](#) \* [pKeyReferenceID](#)
- [uim\\_sessionInformation](#) [sessionInfo](#)
- [uim\\_verifyUIMPIN](#) [verifyPIN](#)
- [uint16\\_t](#) [Tlvresult](#)

### 8.530.1 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>
<i>verifyPIN</i>	<ul style="list-style-type: none"> <li>• See <a href="#">verifyUIMPIN</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>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> <li>Valid Values <ul style="list-style-type: none"> <li>0 - Result of operation in response. Indication will not be generated by the modem</li> <li>Any other positive number - Result of operation in indication. Indication will have same token value set by this function</li> </ul> </li> </ul>

**Note**

Using NULL for the pointers would make sure that the parameter is not added to the request.

**8.530.2 Field Documentation**

8.530.2.1 `uim_encryptedPIN1*` `pack_uim_VerifyPin_t::pEncryptedPIN1`

8.530.2.2 `uint32_t*` `pack_uim_VerifyPin_t::pIndicationToken`

8.530.2.3 `uint8_t*` `pack_uim_VerifyPin_t::pKeyReferenceID`

8.530.2.4 `uim_sessionInformation` `pack_uim_VerifyPin_t::sessionInfo`

8.530.2.5 `uint16_t` `pack_uim_VerifyPin_t::Tlvresult`

8.530.2.6 `uim_verifyUIMPIN` `pack_uim_VerifyPin_t::verifyPIN`

**8.531 `pack_wds_GetDefaultProfile_t` Struct Reference****Data Fields**

- `uint32_t` [profiletype](#)

**8.531.1 Detailed Description****Parameters**

<i>profiletype</i>	profile type
--------------------	--------------

**8.531.2 Field Documentation**

8.531.2.1 `uint32_t` `pack_wds_GetDefaultProfile_t::profiletype`

**8.532 `pack_wds_GetDefaultProfileNum_t` Struct Reference**

## Data Fields

- uint8\_t [type](#)
- uint8\_t [family](#)

### 8.532.1 Detailed Description

#### Parameters

<i>type</i>	profile type <ul style="list-style-type: none"><li>• 0 - 3GPP</li><li>• 1 - 3GPP2</li></ul>
<i>type</i>	profile family <ul style="list-style-type: none"><li>• 0 - Embedded</li><li>• 1 - Tethered</li></ul>

### 8.532.2 Field Documentation

8.532.2.1 uint8\_t pack\_wds\_GetDefaultProfileNum\_t::family

8.532.2.2 uint8\_t pack\_wds\_GetDefaultProfileNum\_t::type

## 8.533 pack\_wds\_GetDormancyState\_t Struct Reference

## 8.534 pack\_wds\_GetLastMobileIPError\_t Struct Reference

## 8.535 pack\_wds\_GetMobileIP\_t Struct Reference

## 8.536 pack\_wds\_GetMobileIPProfile\_t Struct Reference

## Data Fields

- uint8\_t [index](#)

### 8.536.1 Detailed Description

#### Parameters

<i>index</i>	mobile ip profile identifier
--------------	------------------------------

### 8.536.2 Field Documentation

8.536.2.1 uint8\_t pack\_wds\_GetMobileIPProfile\_t::index

## 8.537 pack\_wds\_GetPacketStatus\_t Struct Reference

## Data Fields

- uint32\_t [statmask](#)

### 8.537.1 Detailed Description

#### Parameters

<i>statmask</i>	packet statistics mask
-----------------	------------------------

### 8.537.2 Field Documentation

8.537.2.1 uint32\_t [pack\\_wds\\_GetPacketStatus\\_t::statmask](#)

## 8.538 [pack\\_wds\\_GetSessionDuration\\_t](#) Struct Reference

## 8.539 [pack\\_wds\\_RMSetTransferStatistics\\_t](#) Struct Reference

## Data Fields

- [rmTrasnferStaticsReq](#) [RmTrasnferStaticsReq](#)

### 8.539.1 Detailed Description

#### Parameters

<a href="#">rmTrasnfer-StaticsReq</a>	RM Transfer Statistics Indicator
---------------------------------------	----------------------------------

### 8.539.2 Field Documentation

8.539.2.1 [rmTrasnferStaticsReq](#) [pack\\_wds\\_RMSetTransferStatistics\\_t::RmTrasnferStaticsReq](#)

## 8.540 [pack\\_wds\\_SetDefaultProfile\\_t](#) Struct Reference

## Data Fields

- uint32\_t [profileType](#)
- uint32\_t [pdpType](#)
- uint32\_t [ipAddress](#)
- uint32\_t [primaryDNS](#)
- uint32\_t [secondaryDNS](#)
- uint32\_t [authentication](#)
- uint8\_t \* [pName](#)
- uint8\_t \* [pUsername](#)
- uint8\_t \* [pApnname](#)
- uint8\_t \* [pPassword](#)

### 8.540.1 Detailed Description

## Parameters

<i>profileType</i>	profile type
<i>pdpType</i>	Packet Data Protocol (PDP) type
<i>ipAddress</i>	ip address
<i>primaryDNS</i>	primary dns
<i>secondaryDNS</i>	secondry dns
<i>authentication</i>	authentication type
<i>name</i>	name of the profile
<i>apnname</i>	apn name
<i>username</i>	username of the profile
<i>password</i>	password of profile

## 8.540.2 Field Documentation

8.540.2.1 uint32\_t pack\_wds\_SetDefaultProfile\_t::authentication

8.540.2.2 uint32\_t pack\_wds\_SetDefaultProfile\_t::ipAddress

8.540.2.3 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pApnname

8.540.2.4 uint32\_t pack\_wds\_SetDefaultProfile\_t::pdpType

8.540.2.5 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pName

8.540.2.6 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pPassword

8.540.2.7 uint32\_t pack\_wds\_SetDefaultProfile\_t::primaryDNS

8.540.2.8 uint32\_t pack\_wds\_SetDefaultProfile\_t::profileType

8.540.2.9 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pUsername

8.540.2.10 uint32\_t pack\_wds\_SetDefaultProfile\_t::secondaryDNS

## 8.541 pack\_wds\_SetDefaultProfileNum\_t Struct Reference

## Data Fields

- [uint8\\_t type](#)
- [uint8\\_t family](#)
- [uint8\\_t index](#)

## 8.541.1 Field Documentation

8.541.1.1 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::family

8.541.1.2 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::index

8.541.1.3 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::type

## 8.542 pack\_wds\_SetMobileIPProfile\_t Struct Reference

## Data Fields

- `int8_t` `spc` [10]
- `uint8_t` `index`
- `uint8_t` \* `pEnabled`
- `uint32_t` \* `pAddress`
- `uint32_t` \* `pPrimaryHA`
- `uint32_t` \* `pSecondaryHA`
- `uint8_t` \* `pRevTunneling`
- `int8_t` \* `pNAI`
- `uint32_t` \* `pHASPI`
- `uint32_t` \* `pAAASPI`
- `int8_t` \* `pMNHA`
- `int8_t` \* `pMNAAA`

### 8.542.1 Detailed Description

#### Parameters

<i>spc</i>	service programming code string
<i>index</i>	Index of the profile to modify
<i>pEnabled</i>	Enable profile 0-disable nonzero enable
<i>pAddress</i>	Home IPv4 address
<i>pPrimaryHA</i>	Primary home agent IPv4 address
<i>pSecondaryHA</i>	secondary home agent IPv4 address
<i>pRevTunneling</i>	Enable reverse tunneling 0-disable nonzero enable
<i>pNAI</i>	Network access identifier string
<i>pHASPI</i>	Home agent security parameter index
<i>pAAASPI</i>	AAA server security parameter index
<i>pMNHA</i>	MN-HA key string
<i>pMNAAA</i>	MN-AAA key string

### 8.542.2 Field Documentation

8.542.2.1 `uint8_t` `pack_wds_SetMobileIPProfile_t::index`

8.542.2.2 `uint32_t`\* `pack_wds_SetMobileIPProfile_t::pAAASPI`

8.542.2.3 `uint32_t`\* `pack_wds_SetMobileIPProfile_t::pAddress`

8.542.2.4 `uint8_t`\* `pack_wds_SetMobileIPProfile_t::pEnabled`

8.542.2.5 `uint32_t`\* `pack_wds_SetMobileIPProfile_t::pHASPI`

8.542.2.6 `int8_t`\* `pack_wds_SetMobileIPProfile_t::pMNAAA`

8.542.2.7 `int8_t`\* `pack_wds_SetMobileIPProfile_t::pMNHA`

8.542.2.8 `int8_t`\* `pack_wds_SetMobileIPProfile_t::pNAI`

8.542.2.9 `uint32_t`\* `pack_wds_SetMobileIPProfile_t::pPrimaryHA`

8.542.2.10 `uint8_t`\* `pack_wds_SetMobileIPProfile_t::pRevTunneling`

8.542.2.11 uint32\_t\* pack\_wds\_SetMobilePPProfile\_t::pSecondaryHA

8.542.2.12 int8\_t pack\_wds\_SetMobilePPProfile\_t::spc[10]

## 8.543 pack\_wds\_SLQSCreateProfile\_t Struct Reference

### Data Fields

- uint8\_t \* pProfileId
- uint8\_t \* pProfileType
- wds\_profileInfo \* pCurProfile

### 8.543.1 Detailed Description

#### 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 3GPP and 3GPP2 profile</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

### 8.543.2 Field Documentation

8.543.2.1 wds\_profileInfo\* pack\_wds\_SLQSCreateProfile\_t::pCurProfile

8.543.2.2 uint8\_t\* pack\_wds\_SLQSCreateProfile\_t::pProfileId

8.543.2.3 uint8\_t\* pack\_wds\_SLQSCreateProfile\_t::pProfileType

## 8.544 pack\_wds\_SLQSDestroyProfile\_t Struct Reference

### Data Fields

- uint8\_t profileType
- uint8\_t profileIndex

### 8.544.1 Detailed Description

## Parameters

<i>profileType</i>	profile type
<i>profileIndex</i>	profile index

## 8.544.2 Field Documentation

8.544.2.1 uint8\_t pack\_wds\_SLQSDelProfile\_t::profileIndex

8.544.2.2 uint8\_t pack\_wds\_SLQSDelProfile\_t::profileType

## 8.545 pack\_wds\_SLQSGetCurrDataSystemStat\_t Struct Reference

## 8.546 pack\_wds\_SLQSGetDataBearerTechnology\_t Struct Reference

## 8.547 pack\_wds\_SLQSGetDUNCallInfo\_t Struct Reference

## Data Fields

- uint32\_t [Mask](#)
- uint8\_t \* [pReportConnStatus](#)
- [transferStatInd](#) \* [pTransferStatInd](#)
- uint8\_t \* [pReportDormStatus](#)
- uint8\_t \* [pReportDataBearerTech](#)
- uint8\_t \* [pReportChannelRate](#)

## 8.547.1 Detailed Description

## Parameters

<i>Mask</i>	mask bits corresponding to the information requested to 1
<i>pReportConn- Status</i>	Connect Status Indicator
<i>pTransferStatInd</i>	Transfer Statistics Indicator
<i>pReportDorm- Status</i>	Dormancy Status Indicator
<i>pReportData- BearerTech</i>	Current Data Bearer Technology Indicator
<i>pReport- ChannelRate</i>	Channel Rate Indicator

## 8.547.2 Field Documentation

8.547.2.1 uint32\_t pack\_wds\_SLQSGetDUNCallInfo\_t::Mask

8.547.2.2 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportChannelRate

8.547.2.3 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportConnStatus

8.547.2.4 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDataBearerTech

8.547.2.5 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDormStatus

8.547.2.6 transferStatInd\* pack\_wds\_SLQSGetDUNCallInfo\_t::pTransferStatInd

## 8.548 pack\_wds\_SLQSGetProfileSettings\_t Struct Reference

### Data Fields

- uint8\_t [ProfileId](#)
- uint8\_t [ProfileType](#)

### 8.548.1 Detailed Description

#### 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></ul></li></ul>

#### Note

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

#### Returns

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

#### See Also

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

### 8.548.2 Field Documentation

8.548.2.1 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileId

8.548.2.2 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileType

## 8.549 pack\_wds\_SLQSGetRuntimeSettings\_t Struct Reference

### Data Fields

- uint32\_t \* [pReqSettings](#)

### 8.549.1 Detailed Description

#### Parameters

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

## 8.549.2 Field Documentation

8.549.2.1 `uint32_t* pack_wds_SLQSGetRuntimeSettings_t::pReqSettings`

## 8.550 `pack_wds_SLQSModifyProfile_t` Struct Reference

### Data Fields

- `uint8_t * pProfileId`
- `uint8_t * pProfileType`
- `wds_profileInfo curProfile`

### 8.550.1 Detailed Description

#### 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 3GPP and 3GPP2 profile</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

**8.550.2 Field Documentation**

8.550.2.1 wds\_profileInfo pack\_wds\_SLQSMModifyProfile\_t::curProfile

8.550.2.2 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileId

8.550.2.3 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileType

**8.551 pack\_wds\_SLQSSet3GPPConfigItem\_t Struct Reference****Data Fields**

- uint16\_t [profileList](#) [5]
- uint8\_t [defaultPDNEnabled](#)
- uint8\_t [\\_3gppRelease](#)
- uint16\_t [LTEAttachProfileList](#) [24]
- uint16\_t [LTEAttachProfileListLen](#)

**8.551.1 Detailed Description****Parameters**

<i>profileList</i>	Profile List
<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>0 - disabled</li> <li>1 - enabled</li> </ul>
<i>_3gppRelease</i>	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>

<i>LTEAttach-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> <li>– Please provide attach profiles in order of decreasing priority in this list.</li> </ul> </li> </ul>
<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: 1-24</li> <li>– This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>

### 8.551.2 Field Documentation

8.551.2.1 `uint8_t pack_wds_SLQSSet3GPPConfigItem_t::_3gppRelease`

8.551.2.2 `uint8_t pack_wds_SLQSSet3GPPConfigItem_t::defaultPDNEnabled`

8.551.2.3 `uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileList[24]`

8.551.2.4 `uint16_t pack_wds_SLQSSet3GPPConfigItem_t::LTEAttachProfileListLen`

8.551.2.5 `uint16_t pack_wds_SLQSSet3GPPConfigItem_t::profileList[5]`

## 8.552 `pack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

### Data Fields

- `uint8_t IPFamilyPreference`

### 8.552.1 Detailed Description

#### Parameters

<i>IPFamily-Preference</i>	IP Family preference <ul style="list-style-type: none"> <li>• <code>PACK_WDS_IPV4</code> IP Version 4</li> <li>• <code>PACK_WDS_IPV6</code> IP Version 6</li> </ul>
----------------------------	---

### 8.552.2 Field Documentation

8.552.2.1 `uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference`

## 8.553 `pack_wds_SLQSSetWdsEventCallback_t` Struct Reference

### Data Fields

- `uint8_t dataBearer`
- `uint8_t dormancyStatus`
- `uint8_t mobileIP`

- uint8\_t [transferStats](#)
- uint8\_t [currentDataBearer](#)
- uint8\_t [dataSystemStatus](#)
- uint8\_t [interval](#)

### 8.553.1 Detailed Description

#### Parameters

<i>dataBearer</i>	data bearer
<i>dormancyStatus</i>	dormancy status
<i>mobileIP</i>	mobile IP
<i>currentData-Bearer</i>	current data bearer
<i>dataSystem-Status</i>	data system status
<i>interval</i>	interval

### 8.553.2 Field Documentation

8.553.2.1 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::currentDataBearer

8.553.2.2 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataBearer

8.553.2.3 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataSystemStatus

8.553.2.4 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dormancyStatus

8.553.2.5 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::interval

8.553.2.6 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::mobileIP

8.553.2.7 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::transferStats

## 8.554 pack\_wds\_SLQSSGetDHCPv4ClientConfig\_t Struct Reference

#### Data Fields

- [wdsDhcpv4ProfileId](#) \* [pProfileId](#)

### 8.554.1 Detailed Description

#### Parameters

<i>pProfileId</i>	pointer to Profile Id structure
-------------------	---------------------------------

### 8.554.2 Field Documentation

8.554.2.1 wdsDhcpv4ProfileId\* pack\_wds\_SLQSSGetDHCPv4ClientConfig\_t::pProfileId

## 8.555 pack\_wds\_SLQSSStartDataSession\_t Struct Reference

## Data Fields

- uint8\_t \* [pTech](#)
- uint32\_t \* [pprofileid3gpp](#)
- uint32\_t \* [pprofileid3gpp2](#)
- uint32\_t \* [pAuth](#)
- char \* [pUser](#)
- char \* [pPass](#)

### 8.555.1 Detailed Description

#### Parameters

<i>pTech</i>	<ul style="list-style-type: none"> <li>• Indicates the technology preference <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> <li>• optional</li> </ul>
<i>pprofileid3gpp</i>	<ul style="list-style-type: none"> <li>• pointer to 3GPP profile id</li> <li>• optional</li> </ul>
<i>pprofileid3gpp2</i>	<ul style="list-style-type: none"> <li>• pointer to 3GPPs profile id</li> <li>• optional</li> </ul>
<i>pAuth</i>	<ul style="list-style-type: none"> <li>• Authentication type, it can be PAP or CHAP</li> <li>• optional</li> </ul>
<i>pUser</i>	<ul style="list-style-type: none"> <li>• username for authentication process</li> <li>• optional</li> </ul>
<i>pPass</i>	<ul style="list-style-type: none"> <li>• password for authentication process</li> <li>• optional</li> </ul>

### 8.555.2 Field Documentation

8.555.2.1 uint32\_t\* [pack\\_wds\\_SLQSStartDataSession\\_t::pAuth](#)

8.555.2.2 char\* [pack\\_wds\\_SLQSStartDataSession\\_t::pPass](#)

8.555.2.3 uint32\_t\* [pack\\_wds\\_SLQSStartDataSession\\_t::pprofileid3gpp](#)

8.555.2.4 uint32\_t\* [pack\\_wds\\_SLQSStartDataSession\\_t::pprofileid3gpp2](#)

8.555.2.5 `uint8_t*` `pack_wds_SLQSStartDataSession_t::pTech`

8.555.2.6 `char*` `pack_wds_SLQSStartDataSession_t::pUser`

## 8.556 `pack_wds_SLQSStopDataSession_t` Struct Reference

### Data Fields

- `uint32_t` \* [psid](#)

### 8.556.1 Detailed Description

#### Parameters

<i>sid</i>	session id
------------	------------

### 8.556.2 Field Documentation

8.556.2.1 `uint32_t*` `pack_wds_SLQSStopDataSession_t::psid`

## 8.557 `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t` Struct Reference

### Data Fields

- `uint8_t` [contextId](#)
- `uint8_t` [contextType](#)

### 8.557.1 Detailed Description

#### Parameters

<i>contextId</i>	Context Identifier
<i>contextType</i>	Context Type 0-3GPP 1-3GPP2

### 8.557.2 Field Documentation

8.557.2.1 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.557.2.2 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType`

## 8.558 `PackCreateProfileOut` Struct Reference

### Data Fields

- `uint8_t` [ProfileType](#)
- `uint8_t` [ProfileIndex](#)
- `uint16_t` [ExtErrorCode](#)

### 8.558.1 Field Documentation

8.558.1.1 `uint16_t` `PackCreateProfileOut::ExtErrorCode`

8.558.1.2 `uint8_t PackCreateProfileOut::ProfileIndex`

8.558.1.3 `uint8_t PackCreateProfileOut::ProfileType`

## 8.559 packgetDyingGaspCfg Struct Reference

### Data Fields

- `uint8_t * pDestSMSNum`
- `uint8_t * pDestSMSContent`

### 8.559.1 Detailed Description

#### Parameters

<i>pDestSMSNum</i> [-IN]	<ul style="list-style-type: none"> <li>• SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>• Optional parameter.</li> </ul>
<i>pDestSMSContent</i> [IN]	<ul style="list-style-type: none"> <li>• SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>• Optional parameter.</li> </ul>

### 8.559.2 Field Documentation

8.559.2.1 `uint8_t* packgetDyingGaspCfg::pDestSMSContent`

8.559.2.2 `uint8_t* packgetDyingGaspCfg::pDestSMSNum`

## 8.560 packgetDyingGaspStatistics Struct Reference

### Data Fields

- `uint32_t * pTimeStamp`
- `uint8_t * pSMSAttemptedFlag`

### 8.560.1 Detailed Description

#### Parameters

<i>TimeStamp</i> [OUT]	<ul style="list-style-type: none"> <li>• Time Stamp.</li> </ul>
<i>SMSAttemptedFlag</i> [OUT]	<ul style="list-style-type: none"> <li>• SMS Attempted Flag.</li> </ul>

### 8.560.2 Field Documentation

8.560.2.1 `uint8_t* packgetDyingGaspStatistics::pSMSAttemptedFlag`

8.560.2.2 `uint32_t* packetDyingGaspStatistics::pTimeStamp`

## 8.561 PCMparams Struct Reference

### Data Fields

- [BYTE iFaceTabLen](#)
- [BYTE iFaceTab](#) [255]

#### 8.561.1 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>

#### 8.561.2 Field Documentation

8.561.2.1 **BYTE** PCMparams::iFaceTab[255]

8.561.2.2 **BYTE** PCMparams::iFaceTabLen

## 8.562 PCSCFFQDNAddress Struct Reference

### Data Fields

- [WORD fqdnLen](#)
- [CHAR fqdnAddr](#) [256]

#### 8.562.1 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>

#### 8.562.2 Field Documentation

8.562.2.1 CHAR PCSCFFQDNAddress::fqdnAddr[256]

8.562.2.2 WORD PCSCFFQDNAddress::fqdnLen

## 8.563 PCSCFFQDNAddressList Struct Reference

### Data Fields

- [BYTE numInstances](#)
- struct [PCSCFFQDNAddress pcsfQDNAddress](#) [10]

### 8.563.1 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>pcsfQDN-Address</i>	<ul style="list-style-type: none"> <li>• FQDN address information(Max 10 addresses)</li> </ul>

### 8.563.2 Field Documentation

8.563.2.1 BYTE PCSCFFQDNAddressList::numInstances

8.563.2.2 struct PCSCFFQDNAddress PCSCFFQDNAddressList::pcsfQDNAddress[10]

## 8.564 PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

- [BYTE numInstances](#)
- [ULONG pcsfIPv4Addr](#) [64]

### 8.564.1 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>pcsfIPv4Addr</i>	<ul style="list-style-type: none"> <li>• P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)</li> </ul>

### 8.564.2 Field Documentation

8.564.2.1 BYTE PCSCFIPv4ServerAddressList::numInstances

8.564.2.2 ULONG PCSCFIPv4ServerAddressList::pcscfIPv4Addr[64]

## 8.565 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

### 8.565.1 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 to IEEE 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 pAltitudeWrt-Sealevel 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 pAltitudeWrtEllipsoid 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>

## 8.565.2 Field Documentation

8.565.2.1 **ULONG\*** PDSPositionData::pAltitudeWrtEllipsoid

8.565.2.2 **ULONG\*** PDSPositionData::pAltitudeWrtSealevel

8.565.2.3 **BYTE\*** PDSPositionData::pHorizontalConfidence

8.565.2.4 **ULONG\*** PDSPositionData::pHorizontalUncCircular

8.565.2.5 **ULONGLONG\*** PDSPositionData::pLatitude

8.565.2.6 **ULONGLONG\*** PDSPositionData::pLongitude

8.565.2.7 **BYTE\*** PDSPositionData::pPositionSource

8.565.2.8 **ULONGLONG\*** PDSPositionData::pTimeStamp

8.565.2.9 **BYTE\*** PDSPositionData::pTimeType

8.565.2.10 **BYTE\*** PDSPositionData::pVerticalConfidence

8.565.2.11 ULONG\* PDSPositionData::pVerticalUnc

## 8.566 PDSPosMethodStateReq Struct Reference

### Data Fields

- [BYTE \\* pXtraTimeState](#)
- [BYTE \\* pXtraDataState](#)
- [BYTE \\* pWifiState](#)

### 8.566.1 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>

### 8.566.2 Field Documentation

8.566.2.1 BYTE\* PDSPosMethodStateReq::pWifiState

8.566.2.2 BYTE\* PDSPosMethodStateReq::pXtraDataState

8.566.2.3 BYTE\* PDSPosMethodStateReq::pXtraTimeState

## 8.567 peerNumberInfo Struct Reference

### Data Fields

- [BYTE callID](#)
- [BYTE numPI](#)
- [BYTE numSI](#)
- [BYTE numType](#)
- [BYTE numPlan](#)

- BYTE numLen
- BYTE number [81]

### 8.567.1 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</i> [MAX_C-ALL_NO_LEN]	<ul style="list-style-type: none"><li>number of numLen length, NULL terminated.</li></ul>

## 8.567.2 Field Documentation

8.567.2.1 **BYTE** peerNumberInfo::callID

8.567.2.2 **BYTE** peerNumberInfo::number[81]

8.567.2.3 **BYTE** peerNumberInfo::numLen

8.567.2.4 **BYTE** peerNumberInfo::numPI

8.567.2.5 **BYTE** peerNumberInfo::numPlan

8.567.2.6 **BYTE** peerNumberInfo::numSI

8.567.2.7 **BYTE** peerNumberInfo::numType

## 8.568 personalizationStatus Struct Reference

### Data Fields

- [BYTE numFeatures](#)
- [BYTE feature](#) [12]
- [BYTE verifyLeft](#) [12]
- [BYTE unblockLeft](#) [12]

### 8.568.1 Detailed Description

This structure contains the information about the card result.

## Parameters

<i>numFeatures</i>	<ul style="list-style-type: none"> <li>Number of active personalization features. The following block is repeated for each feature.</li> </ul>
<i>feature</i>	<ul style="list-style-type: none"> <li>Indicates the personalization feature to deactivate or unblock. Valid values: <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>verifyLeft</i>	<ul style="list-style-type: none"> <li>Number of the remaining attempts to verify the personalization feature.</li> </ul>
<i>unblockLeft</i>	<ul style="list-style-type: none"> <li>Number of the remaining attempts to unblock the personalization feature.</li> </ul>

## 8.568.2 Field Documentation

8.568.2.1 BYTE personalizationStatus::feature[12]

8.568.2.2 BYTE personalizationStatus::numFeatures

8.568.2.3 BYTE personalizationStatus::unblockLeft[12]

8.568.2.4 BYTE personalizationStatus::verifyLeft[12]

## 8.569 PhyCaAggPcellInfo Struct Reference

## Data Fields

- int [pci](#)
- int [freq](#)
- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) [dl\\_bw\\_value](#)
- int [iLTEbandValue](#)
- BYTE [TlvPresent](#)

## 8.569.1 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>

## 8.569.2 Field Documentation

8.569.2.1 NAS\_LTE\_CPHY\_CA\_BW\_NRB PhyCaAggPcellInfo::dl\_bw\_value

8.569.2.2 int PhyCaAggPcellInfo::freq

8.569.2.3 int PhyCaAggPcellInfo::iLTEbandValue

8.569.2.4 int PhyCaAggPcellInfo::pci

8.569.2.5 BYTE PhyCaAggPcellInfo::TlvPresent

## 8.570 PhyCaAggScellIDBw Struct Reference

## Data Fields

- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB dl\\_bw\\_value](#)
- [BYTE TlvPresent](#)

## 8.570.1 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>
--------------------	--

## 8.570.2 Field Documentation

8.570.2.1 NAS\_LTE\_CPHY\_CA\_BW\_NRB PhyCaAggScellIDBw::dl\_bw\_value

8.570.2.2 BYTE PhyCaAggScellIDBw::TlvPresent

## 8.571 PhyCaAggScellIndex Struct Reference

### Data Fields

- [BYTE scell\\_idx](#)
- [BYTE TlvPresent](#)

### 8.571.1 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>

### 8.571.2 Field Documentation

8.571.2.1 BYTE PhyCaAggScellIndex::scell\_idx

8.571.2.2 BYTE PhyCaAggScellIndex::TlvPresent

## 8.572 PhyCaAggScellIndType Struct Reference

### Data Fields

- int [pci](#)
- int [freq](#)
- [NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) [scell\\_state](#)
- [BYTE TlvPresent](#)

### 8.572.1 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>

## 8.572.2 Field Documentation

8.572.2.1 int PhyCaAggScellIndType::freq

8.572.2.2 int PhyCaAggScellIndType::pci

8.572.2.3 NAS\_LTE\_CPHY\_SCELL\_STATE PhyCaAggScellIndType::scell\_state

8.572.2.4 BYTE PhyCaAggScellIndType::TlvPresent

## 8.573 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

### 8.573.1 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. <ul style="list-style-type: none"> <li>– 120 - LTE E-UTRA Operating Band 1</li> <li>– 121 - LTE E-UTRA Operating Band 2</li> <li>– 122 - LTE E-UTRA Operating Band 3</li> <li>– 123 - LTE E-UTRA Operating Band 4</li> <li>– 124 - LTE E-UTRA Operating Band 5</li> <li>– 125 - LTE E-UTRA Operating Band 6</li> <li>– 126 - LTE E-UTRA Operating Band 7</li> <li>– 127 - LTE E-UTRA Operating Band 8</li> <li>– 128 - LTE E-UTRA Operating Band 9</li> <li>– 129 - LTE E-UTRA Operating Band 10</li> <li>– 130 - LTE E-UTRA Operating Band 11</li> <li>– 131 - LTE E-UTRA Operating Band 12</li> <li>– 132 - LTE E-UTRA Operating Band 13</li> <li>– 133 - LTE E-UTRA Operating Band 14</li> <li>– 134 - LTE E-UTRA Operating Band 17</li> <li>– 135 - LTE E-UTRA Operating Band 33</li> <li>– 136 - LTE E-UTRA Operating Band 34</li> <li>– 137 - LTE E-UTRA Operating Band 35</li> <li>– 138 - LTE E-UTRA Operating Band 36</li> <li>– 139 - LTE E-UTRA Operating Band 37</li> <li>– 140 - LTE E-UTRA Operating Band 38</li> <li>– 141 - LTE E-UTRA Operating Band 39</li> <li>– 142 - LTE E-UTRA Operating Band 40</li> <li>– 143 - LTE E-UTRA Operating Band 18</li> <li>– 144 - LTE E-UTRA Operating Band 19</li> <li>– 145 - LTE E-UTRA Operating Band 20</li> <li>– 146 - LTE E-UTRA Operating Band 21</li> <li>– 147 - LTE E-UTRA Operating Band 24</li> <li>– 148 - LTE E-UTRA Operating Band 25</li> <li>– 149 - LTE E-UTRA Operating Band 41</li> <li>– 150 - LTE E-UTRA Operating Band 42</li> <li>– 151 - LTE E-UTRA Operating Band 43</li> <li>– 152 - LTE E-UTRA Operating Band 23</li> <li>– 153 - LTE E-UTRA Operating Band 26</li> <li>– 154 - LTE E-UTRA Operating Band 32</li> <li>– 155 - LTE E-UTRA Operating Band 125</li> <li>– 156 - LTE E-UTRA Operating Band 126</li> <li>– 157 - LTE E-UTRA Operating Band 127</li> <li>– 158 - LTE E-UTRA Operating Band 28</li> <li>– 159 - LTE E-UTRA Operating Band 29</li> <li>– 160 - LTE E-UTRA Operating Band 30</li> </ul> </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>
-------------------	--

## 8.573.2 Field Documentation

8.573.2.1 **NAS\_LTE\_CPHY\_CA\_BW\_NRB** `PhyCaAggScellInfo::dl_bw_value`

8.573.2.2 `int` `PhyCaAggScellInfo::freq`

8.573.2.3 `int` `PhyCaAggScellInfo::ltebandValue`

8.573.2.4 `int` `PhyCaAggScellInfo::pci`

8.573.2.5 **NAS\_LTE\_CPHY\_SCELL\_STATE** `PhyCaAggScellInfo::scell_state`

8.573.2.6 **BYTE** `PhyCaAggScellInfo::TlvPresent`

## 8.574 PilotSetData Struct Reference

### Data Fields

- [BYTE](#) `NumPilots`
- [PilotSetParams](#) \* `pPilotSetInfo`

### 8.574.1 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.

## 8.574.2 Field Documentation

8.574.2.1 **BYTE** `PilotSetData::NumPilots`

8.574.2.2 **PilotSetParams**\* `PilotSetData::pPilotSetInfo`

## 8.575 PilotSetParams Struct Reference

### Data Fields

- [ULONG PilotType](#)
- [WORD PilotPN](#)
- [WORD PilotStrength](#)

### 8.575.1 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>

### 8.575.2 Field Documentation

8.575.2.1 **WORD** PilotSetParams::PilotPN

8.575.2.2 **WORD** PilotSetParams::PilotStrength

8.575.2.3 **ULONG** PilotSetParams::PilotType

## 8.576 pktErrRate Struct Reference

### Data Fields

- [WORD multiplier](#)
- [WORD exponent](#)

### 8.576.1 Detailed Description

This structure contains the IP flow packet error rate

#### Parameters

<i>multiplier</i>	Factor m in calculating packet error rate: $E = m * 10^{**}(-p)$
<i>exponent</i>	Factor p in calculating packet error rate (see above)

### 8.576.2 Field Documentation

8.576.2.1 **WORD** pktErrRate::exponent

8.576.2.2 WORD pktErrRate::multiplier

## 8.577 PLMNNetworkName Struct Reference

### Data Fields

- [BYTE numInstance](#)
- [PLMNNetworkNameData PLMNNetName](#) [255]

### 8.577.1 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>

### 8.577.2 Field Documentation

8.577.2.1 BYTE PLMNNetworkName::numInstance

8.577.2.2 PLMNNetworkNameData PLMNNetworkName::PLMNNetName[255]

## 8.578 PLMNNetworkNameData Struct Reference

### Data Fields

- [BYTE codingScheme](#)
- [BYTE countryInitials](#)
- [BYTE longNameSpareBits](#)
- [BYTE shortNameSpareBits](#)
- [BYTE longNameLen](#)
- [BYTE longName](#) [255]
- [BYTE shortNameLen](#)
- [BYTE shortName](#) [255]

### 8.578.1 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>
<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>

## 8.578.2 Field Documentation

### 8.578.2.1 BYTE PLMNNetworkNameData::codingScheme

### 8.578.2.2 BYTE PLMNNetworkNameData::countryInitials

### 8.578.2.3 BYTE PLMNNetworkNameData::longName[255]

8.578.2.4 BYTE PLMNNetworkNameData::longNameLen

8.578.2.5 BYTE PLMNNetworkNameData::longNameSpareBits

8.578.2.6 BYTE PLMNNetworkNameData::shortName[255]

8.578.2.7 BYTE PLMNNetworkNameData::shortNameLen

8.578.2.8 BYTE PLMNNetworkNameData::shortNameSpareBits

## 8.579 Port Struct Reference

### Data Fields

- [WORD port](#)
- [WORD range](#)

### 8.579.1 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

### 8.579.2 Field Documentation

8.579.2.1 WORD Port::port

8.579.2.2 WORD Port::range

## 8.580 precisionDilution\_s Struct Reference

### Data Fields

- [ULONG PDOP](#)
- [ULONG HDOP](#)
- [ULONG VDOP](#)

### 8.580.1 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>• <math>PDOP = \text{square root of (Square of HDOP + Square of VDOP}^2 \text{ )}</math></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>

## 8.580.2 Field Documentation

8.580.2.1 **ULONG** precisionDilution\_s::HDOP

8.580.2.2 **ULONG** precisionDilution\_s::PDOP

8.580.2.3 **ULONG** precisionDilution\_s::VDOP

## 8.581 PrefImageList Struct Reference

### Data Fields

- [BYTE](#) listSize
- struct [ImageElement](#) listEntries [2]

### 8.581.1 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>listEntries</i>	<ul style="list-style-type: none"> <li>Array of Image entries( Max array size 2 )</li> <li>See <a href="#">ImageElement</a></li> </ul>

## 8.581.2 Field Documentation

8.581.2.1 **struct ImageElement** PrefImageList::listEntries[2]

8.581.2.2 **BYTE** PrefImageList::listSize

## 8.582 prefVoiceSO Struct Reference

### Data Fields

- [BYTE](#) namID
- [BYTE](#) evrcCapability
- [WORD](#) homePageVoiceSO

- [WORD homeOrigVoiceSO](#)
- [WORD roamOrigVoiceSO](#)

### 8.582.1 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>homePageVoiceSO</i>	<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>homeOrigVoiceSO</i>	<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>

## 8.582.2 Field Documentation

8.582.2.1 **BYTE** prefVoiceSO::evrcCapability

8.582.2.2 **WORD** prefVoiceSO::homeOrigVoiceSO

8.582.2.3 **WORD** prefVoiceSO::homePageVoiceSO

8.582.2.4 **BYTE** prefVoiceSO::namID

8.582.2.5 **WORD** prefVoiceSO::roamOrigVoiceSO

## 8.583 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 \* pGPRSMMinimumQoS
- 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

### 8.583.1 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 pProfileName 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 pAPNName 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="#">SLQSGetProfileSettings()</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="#">SLQSGetProfileSettings()</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="#">SLQSGetProfileSettings()</a>.</li> </ul>

## 8.583.2 Field Documentation

8.583.2.1 **BYTE\*** Profile3GPP::pAddrAllocPref

8.583.2.2 **BYTE\*** Profile3GPP::pAPNClass

8.583.2.3 **BYTE\*** Profile3GPP::pAPNDisabledFlag

8.583.2.4 **CHAR\*** Profile3GPP::pAPNName

8.583.2.5 **WORD\*** Profile3GPP::pAPNnameSize

8.583.2.6 **BYTE\*** Profile3GPP::pAuthenticationPref

8.583.2.7 **struct GPRSRequestedQoS\*** Profile3GPP::pGPRSMinimumQoS

8.583.2.8 **struct GPRSRequestedQoS\*** Profile3GPP::pGPRSRequestedQoS

8.583.2.9 **BYTE\*** Profile3GPP::plmCnFlag

8.583.2.10 **ULONG\*** Profile3GPP::pIPv4AddrPref

8.583.2.11 **USHORT\*** Profile3GPP::pIPv6AddPref

8.583.2.12 **CHAR\*** Profile3GPP::pPassword

8.583.2.13 **WORD\*** Profile3GPP::pPasswordSize

8.583.2.14 **BYTE\*** Profile3GPP::pPcscfAddrUsingDhcp

8.583.2.15 **BYTE\*** Profile3GPP::pPcscfAddrUsingPCO

8.583.2.16 **ULONG\*** Profile3GPP::pPDNInactivTimeout

8.583.2.17 **BYTE\*** Profile3GPP::pPdpAccessConFlag

8.583.2.18 **BYTE\*** Profile3GPP::pPdpContext

- 8.583.2.19 **BYTE\*** Profile3GPP::pPdpDataCompType
- 8.583.2.20 **BYTE\*** Profile3GPP::pPdpHdrCompType
- 8.583.2.21 **BYTE\*** Profile3GPP::pPDPtype
- 8.583.2.22 **ULONG\*** Profile3GPP::pPriDNSIPv4AddPref
- 8.583.2.23 **USHORT\*** Profile3GPP::pPriDNSIPv6addpref
- 8.583.2.24 **BYTE\*** Profile3GPP::pPrimaryID
- 8.583.2.25 **CHAR\*** Profile3GPP::pProfilename
- 8.583.2.26 **WORD\*** Profile3GPP::pProfilenameSize
- 8.583.2.27 **struct QosClassID\*** Profile3GPP::pQosClassID
- 8.583.2.28 **ULONG\*** Profile3GPP::pSecDNSIPv4AddPref
- 8.583.2.29 **USHORT\*** Profile3GPP::pSecDNSIPv6addpref
- 8.583.2.30 **BYTE\*** Profile3GPP::pSecondaryFlag
- 8.583.2.31 **struct TFTIDParams\*** Profile3GPP::pTFTID1Params
- 8.583.2.32 **struct TFTIDParams\*** Profile3GPP::pTFTID2Params
- 8.583.2.33 **struct UMTSQoS\*** Profile3GPP::pUMTSMinQoS
- 8.583.2.34 **struct UMTSReqQoSSigInd\*** Profile3GPP::pUMTSMinQoSsigInd
- 8.583.2.35 **struct UMTSQoS\*** Profile3GPP::pUMTSReqQoS
- 8.583.2.36 **struct UMTSReqQoSSigInd\*** Profile3GPP::pUMTSReqQoSSigInd
- 8.583.2.37 **CHAR\*** Profile3GPP::pUsername
- 8.583.2.38 **WORD\*** Profile3GPP::pUsernameSize

## 8.584 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 \* plsPcscfAddressNedded
- ULONG \* pPrimaryV4DnsAddress
- ULONG \* pSecondaryV4DnsAddress
- USHORT \* pPriV6DnsAddress
- USHORT \* pSecV6DnsAddress
- BYTE \* pRATType
- BYTE \* pAPNEnabled3GPP2
- ULONG \* pPDNInactivTimeout3GPP2
- BYTE \* pAPNClass3GPP2

### 8.584.1 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>pIpcpAck-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>pIpcpCreqRetry-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="#">SLQSGetProfileSettings()</a>.</li> </ul>
<i>pAPNEnabled3GPP2</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="#">SLQSGetProfileSettings()</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="#">SLQSGetProfileSettings()</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>

## 8.584.2 Field Documentation

8.584.2.1 **BYTE\*** Profile3GPP2::pAllowLinger

8.584.2.2 **BYTE\*** Profile3GPP2::pAPNClass3GPP2

8.584.2.3 **BYTE\*** Profile3GPP2::pAPNEnabled3GPP2

8.584.2.4 **CHAR\*** Profile3GPP2::pApnString

8.584.2.5 **WORD\*** Profile3GPP2::pApnStringSize

8.584.2.6 **BYTE\*** Profile3GPP2::pAppPriority

8.584.2.7 **ULONG\*** Profile3GPP2::pAppType

- 8.584.2.8 **CHAR\*** Profile3GPP2::pAuthPassword
- 8.584.2.9 **WORD\*** Profile3GPP2::pAuthPasswordSize
- 8.584.2.10 **BYTE\*** Profile3GPP2::pAuthProtocol
- 8.584.2.11 **BYTE\*** Profile3GPP2::pAuthRetryCount
- 8.584.2.12 **USHORT\*** Profile3GPP2::pAuthTimeout
- 8.584.2.13 **BYTE\*** Profile3GPP2::pDataMode
- 8.584.2.14 **BYTE\*** Profile3GPP2::pDataRate
- 8.584.2.15 **USHORT\*** Profile3GPP2::plpcpAckTimeout
- 8.584.2.16 **BYTE\*** Profile3GPP2::plpcpCreqRetryCount
- 8.584.2.17 **BYTE\*** Profile3GPP2::plsPcscfAddressNedded
- 8.584.2.18 **USHORT\*** Profile3GPP2::pLcpAckTimeout
- 8.584.2.19 **BYTE\*** Profile3GPP2::pLcpCreqRetryCount
- 8.584.2.20 **BYTE\*** Profile3GPP2::pNegoDnsSrvrPref
- 8.584.2.21 **ULONG\*** Profile3GPP2::pPDNInactivTimeout3GPP2
- 8.584.2.22 **BYTE\*** Profile3GPP2::pPdnType
- 8.584.2.23 **ULONG\*** Profile3GPP2::pPppSessCloseTimer1x
- 8.584.2.24 **ULONG\*** Profile3GPP2::pPppSessCloseTimerDO
- 8.584.2.25 **ULONG\*** Profile3GPP2::pPrimaryV4DnsAddress
- 8.584.2.26 **USHORT\*** Profile3GPP2::pPriV6DnsAddress
- 8.584.2.27 **BYTE\*** Profile3GPP2::pRATType
- 8.584.2.28 **ULONG\*** Profile3GPP2::pSecondaryV4DnsAddress
- 8.584.2.29 **USHORT\*** Profile3GPP2::pSecV6DnsAddress
- 8.584.2.30 **CHAR\*** Profile3GPP2::pUserId
- 8.584.2.31 **WORD\*** Profile3GPP2::pUserIdSize

## 8.585 ProfileIdentifier Struct Reference

### Data Fields

- [BYTE profileType](#)
- [BYTE profileIndex](#)

### 8.585.1 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>

### 8.585.2 Field Documentation

8.585.2.1 **BYTE** ProfileIdentifier::profileIndex

8.585.2.2 **BYTE** ProfileIdentifier::profileType

## 8.586 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](#)

### 8.586.1 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>

## 8.586.2 Field Documentation

8.586.2.1 WORD protocolSubtypeElement::AccessMac

8.586.2.2 WORD protocolSubtypeElement::AuthProt

8.586.2.3 WORD protocolSubtypeElement::ControlMac

8.586.2.4 WORD protocolSubtypeElement::EncryptProt

8.586.2.5 WORD protocolSubtypeElement::ForwardMac

8.586.2.6 WORD protocolSubtypeElement::IdleState

8.586.2.7 WORD protocolSubtypeElement::KeyExchange

8.586.2.8 WORD protocolSubtypeElement::MultDisc

8.586.2.9 WORD protocolSubtypeElement::PhysicalLayer

8.586.2.10 WORD protocolSubtypeElement::ReverseMac

8.586.2.11 WORD protocolSubtypeElement::SecProt

8.586.2.12 WORD protocolSubtypeElement::VirtStream

## 8.587 PSDetachReq Struct Reference

### Data Fields

- [BYTE](#) \* [pDetachAction](#)

### 8.587.1 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>
--------------------------------	---

### 8.587.2 Field Documentation

8.587.2.1 [BYTE](#)\* PSDetachReq::pDetachAction

## 8.588 qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- [BYTE](#) [leapSeconds](#)
- [BYTE](#) [localTimeOffset](#)
- [BYTE](#) [daylightSavings](#)

### 8.588.1 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>

## 8.588.2 Field Documentation

8.588.2.1 **BYTE** qaQmi3Gpp2TimeZone::daylightSavings

8.588.2.2 **BYTE** qaQmi3Gpp2TimeZone::leapSeconds

8.588.2.3 **BYTE** qaQmi3Gpp2TimeZone::localTimeOffset

## 8.589 qaQmiInterfaceInfo Struct Reference

### Data Fields

- [BYTE](#) qaQmiinstanceid
- [eQaQMIService](#) qaQmisvctype
- [ULONG](#) v4sessionId
- [ULONG](#) v6sessionId

### 8.589.1 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>

## 8.589.2 Field Documentation

8.589.2.1 **BYTE** qaQmiInterfaceInfo::qaQmiinstanceid

8.589.2.2 **eQaQMIService** qaQmiInterfaceInfo::qaQmisvctype

8.589.2.3 **ULONG** qaQmiInterfaceInfo::v4sessionId

8.589.2.4 **ULONG** qaQmiInterfaceInfo::v6sessionId

## 8.590 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](#) [DTMInd](#)
- [detailSvcInfo](#) [DetailedSvcInfo](#)
- [CDMASysInfoExt](#) [CDMASystemInfoExt](#)
- [BYTE](#) [hdrPersonality](#)
- [WORD](#) [trackAreaCode](#)
- [callBarStatus](#) [CallBarStatus](#)

### 8.590.1 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>
<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>

## 8.590.2 Field Documentation

### 8.590.2.1 WORD qaQmiServingSystemParam::BasestationID

### 8.590.2.2 ULONG qaQmiServingSystemParam::BasestationLatitude

### 8.590.2.3 ULONG qaQmiServingSystemParam::BasestationLongitude

- 8.590.2.4 `callBarStatus` `qaQmiServingSystemParam::CallBarStatus`
- 8.590.2.5 `BYTE` `qaQmiServingSystemParam::CDMA_P_Rev`
- 8.590.2.6 `CDMASysInfoExt` `qaQmiServingSystemParam::CDMASystemInfoExt`
- 8.590.2.7 `ULONG` `qaQmiServingSystemParam::CellID`
- 8.590.2.8 `BYTE` `qaQmiServingSystemParam::concSvcInfo`
- 8.590.2.9 `currentPLMN` `qaQmiServingSystemParam::CurrentPLMN`
- 8.590.2.10 `dataSrvCapabilities` `qaQmiServingSystemParam::DataSrvCapabilities`
- 8.590.2.11 `BYTE` `qaQmiServingSystemParam::defaultRoamInd`
- 8.590.2.12 `detailSvcInfo` `qaQmiServingSystemParam::DetailedSvcInfo`
- 8.590.2.13 `BYTE` `qaQmiServingSystemParam::DTMInd`
- 8.590.2.14 `qaQmi3Gpp2TimeZone` `qaQmiServingSystemParam::Gpp2TimeZone`
- 8.590.2.15 `BYTE` `qaQmiServingSystemParam::GppNetworkDSTAdjustment`
- 8.590.2.16 `BYTE` `qaQmiServingSystemParam::GppTimeZone`
- 8.590.2.17 `BYTE` `qaQmiServingSystemParam::hdrPersonality`
- 8.590.2.18 `WORD` `qaQmiServingSystemParam::Lac`
- 8.590.2.19 `WORD` `qaQmiServingSystemParam::NetworkID`
- 8.590.2.20 `BYTE` `qaQmiServingSystemParam::PRLInd`
- 8.590.2.21 `BYTE` `qaQmiServingSystemParam::roamIndicatorVal`
- 8.590.2.22 `roamIndList` `qaQmiServingSystemParam::RoamingIndicatorList`
- 8.590.2.23 `servSystem` `qaQmiServingSystemParam::ServingSystem`
- 8.590.2.24 `WORD` `qaQmiServingSystemParam::SystemID`
- 8.590.2.25 `WORD` `qaQmiServingSystemParam::trackAreaCode`

## 8.591 QmiCbkCatEventStatusReportInd Struct Reference

### Data Fields

- [BYTE](#) `event_Index`
- struct [CatCommonEventTlv](#) `CCETlv` [11]

### 8.591.1 Field Documentation

- 8.591.1.1 struct `CatCommonEventTlv` `QmiCbkCatEventStatusReportInd::CCETlv`[11]

8.591.1.2 BYTE QmiCbkCatEventStatusReportInd::event\_Index

## 8.592 QmiCbkLocBestAvailPosInd Struct Reference

### Data Fields

- ULONG status
- ULONG \* pXid
- ULONGLONG \* pLatitude
- ULONGLONG \* pLongitude
- ULONG \* pHorUncCircular
- ULONG \* pAltitudeWrtEllipsoid
- ULONG \* pVertUnc
- ULONGLONG \* pTimestampUtc
- ULONG \* pTimeUnc
- ULONG \* pHorUncEllipseSemiMinor
- ULONG \* pHorUncEllipseSemiMajor
- ULONG \* pHorUncEllipseOrientAzimuth
- BYTE \* pHorCirConf
- BYTE \* pHorEllpConf
- ULONG \* pHorReliability
- ULONG \* pSpeedHorizontal
- ULONG \* pSpeedUnc
- ULONG \* pAltitudeWrtMeanSeaLevel
- BYTE \* pVertConfidence
- ULONG \* pVertReliability
- ULONG \* pSpeedVertical
- ULONG \* pSpeedVerticalUnc
- ULONG \* pHeading
- ULONG \* pHeadingUnc
- ULONG \* pMagneticDeviation
- ULONG \* pTechnologyMask
- precisionDilution \* pPrecisionDilution
- gpsTime \* pGpsTime
- ULONG \* pTimeSrc
- sensorDataUsage \* pSensorDataUsage
- svUsedforFix \* pSvUsedforFix

### 8.592.1 Detailed Description

This structure contains Best Available Position

## Parameters

<i>status</i>	<ul style="list-style-type: none"> <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>
<i>xid</i>	Transaction ID that was specified in the Get Best Available Position request.
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>Latitude (specified in WGS84 datum)</li> <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>Longitude (specified in WGS84 datum)</li> <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>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> <li>Altitude With Respect to WGS84 Ellipsoid.</li> <li>Units - Meters</li> <li>Range -500 to 15883</li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>Vertical uncertainty.</li> <li>Units - Meters</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>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</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>pHorCirConf</i>	<ul style="list-style-type: none"> <li>• Horizontal circular uncertainty confidence</li> <li>• Units: Precent</li> <li>• Range: 0 to 99</li> </ul>
<i>pHorEllpConf</i>	<ul style="list-style-type: none"> <li>• Horizontal elliptical uncertainty confidence</li> <li>• Units: Precent</li> <li>• Range: 0 to 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-MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Sea Level.</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>pSpeedVertical-Unc</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>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">gpsTime</a> for more information</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> <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> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </li> </ul>
<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">sensorDataUsage</a> for more information</li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">svUsedforFix</a> for more information</li> </ul>

## 8.592.2 Field Documentation

8.592.2.1 **ULONG\*** QmiCbkLocBestAvailPosInd::pAltitudeWrtEllipsoid

8.592.2.2 **ULONG\*** QmiCbkLocBestAvailPosInd::pAltitudeWrtMeanSeaLevel

8.592.2.3 **gpsTime\*** QmiCbkLocBestAvailPosInd::pGpsTime

8.592.2.4 **ULONG\*** QmiCbkLocBestAvailPosInd::pHeading

8.592.2.5 **ULONG\*** QmiCbkLocBestAvailPosInd::pHeadingUnc

8.592.2.6 **BYTE\*** QmiCbkLocBestAvailPosInd::pHorCirConf

8.592.2.7 **BYTE\*** QmiCbkLocBestAvailPosInd::pHorEllpConf

8.592.2.8 **ULONG\*** QmiCbkLocBestAvailPosInd::pHorReliability

8.592.2.9 **ULONG\*** QmiCbkLocBestAvailPosInd::pHorUncCircular

8.592.2.10 **ULONG\*** QmiCbkLocBestAvailPosInd::pHorUncEllipseOrientAzimuth

8.592.2.11 **ULONG\*** QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMajor

- 8.592.2.12 **ULONG\*** QmiCbkLocBestAvailPosInd::pHorUncEllipseSemiMinor
- 8.592.2.13 **ULONGLONG\*** QmiCbkLocBestAvailPosInd::pLatitude
- 8.592.2.14 **ULONGLONG\*** QmiCbkLocBestAvailPosInd::pLongitude
- 8.592.2.15 **ULONG\*** QmiCbkLocBestAvailPosInd::pMagneticDeviation
- 8.592.2.16 **precisionDilution\*** QmiCbkLocBestAvailPosInd::pPrecisionDilution
- 8.592.2.17 **sensorDataUsage\*** QmiCbkLocBestAvailPosInd::pSensorDataUsage
- 8.592.2.18 **ULONG\*** QmiCbkLocBestAvailPosInd::pSpeedHorizontal
- 8.592.2.19 **ULONG\*** QmiCbkLocBestAvailPosInd::pSpeedUnc
- 8.592.2.20 **ULONG\*** QmiCbkLocBestAvailPosInd::pSpeedVertical
- 8.592.2.21 **ULONG\*** QmiCbkLocBestAvailPosInd::pSpeedVerticalUnc
- 8.592.2.22 **svUsedforFix\*** QmiCbkLocBestAvailPosInd::pSvUsedforFix
- 8.592.2.23 **ULONG\*** QmiCbkLocBestAvailPosInd::pTechnologyMask
- 8.592.2.24 **ULONG\*** QmiCbkLocBestAvailPosInd::pTimeSrc
- 8.592.2.25 **ULONGLONG\*** QmiCbkLocBestAvailPosInd::pTimestampUtc
- 8.592.2.26 **ULONG\*** QmiCbkLocBestAvailPosInd::pTimeUnc
- 8.592.2.27 **BYTE\*** QmiCbkLocBestAvailPosInd::pVertConfidence
- 8.592.2.28 **ULONG\*** QmiCbkLocBestAvailPosInd::pVertReliability
- 8.592.2.29 **ULONG\*** QmiCbkLocBestAvailPosInd::pVertUnc
- 8.592.2.30 **ULONG\*** QmiCbkLocBestAvailPosInd::pXid
- 8.592.2.31 **ULONG** QmiCbkLocBestAvailPosInd::status

## 8.593 QmiCbkLocCradleMountInd Struct Reference

### Data Fields

- [ULONG cradleMountConfigStatus](#)

### 8.593.1 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>
---------------------------------	---

## 8.593.2 Field Documentation

8.593.2.1 ULONG QmiCbkLocCradleMountInd::cradleMountConfigStatus

## 8.594 QmiCbkLocEngineStateInd Struct Reference

## Data Fields

- [ULONG engineState](#)

## 8.594.1 Detailed Description

This structure contains LOC Engine State field.

## Parameters

<i>engineState</i>	<ul style="list-style-type: none"> <li>• Location engine state.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 1 - Location engine is on</li> <li>– 2 - Location engine is off</li> </ul> </li> </ul>
--------------------	---

## 8.594.2 Field Documentation

8.594.2.1 ULONG QmiCbkLocEngineStateInd::engineState

## 8.595 QmiCbkLocEventTimeSyncInd Struct Reference

## Data Fields

- [ULONG timeSyncRefCounter](#)

### 8.595.1 Detailed Description

This structure contains LOC Event Time Sync Reference COUNTER

#### Parameters

<i>timeSyncRefCounter</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>
---------------------------	--

### 8.595.2 Field Documentation

8.595.2.1 **ULONG** QmiCbkLocEventTimeSyncInd::timeSyncRefCounter

## 8.596 QmiCbkLocInjectPositionInd Struct Reference

#### Data Fields

- [ULONG status](#)

### 8.596.1 Detailed Description

Contain the parameters passed for SetLocInjectPositionCallback by the device.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>UTC Position Injection Status</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

### 8.596.2 Field Documentation

8.596.2.1 ULONG QmiCbkLocInjectPositionInd::status

## 8.597 QmiCbkLocInjectSensorDataInd Struct Reference

### Data Fields

- [ULONG injectSensorDataStatus](#)
- [ULONG \\* pOpaqueIdentifier](#)
- [BYTE \\* pAccelSamplesAccepted](#)
- [BYTE \\* pGyroSamplesAccepted](#)
- [BYTE \\* pAccelTempSamplesAccepted](#)
- [BYTE \\* pGyroTempSamplesAccepted](#)

### 8.597.1 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>

### 8.597.2 Field Documentation

8.597.2.1 **ULONG** QmiCbkLocInjectSensorDataInd::injectSensorDataStatus

8.597.2.2 **BYTE\*** QmiCbkLocInjectSensorDataInd::pAccelSamplesAccepted

8.597.2.3 **BYTE\*** QmiCbkLocInjectSensorDataInd::pAccelTempSamplesAccepted

8.597.2.4 **BYTE\*** QmiCbkLocInjectSensorDataInd::pGyroSamplesAccepted

8.597.2.5 **BYTE\*** QmiCbkLocInjectSensorDataInd::pGyroTempSamplesAccepted

8.597.2.6 **ULONG\*** QmiCbkLocInjectSensorDataInd::pOpaqueIdentifier

## 8.598 QmiCbkLocInjectTimeInd Struct Reference

### Data Fields

- [ULONG injectTimeSyncStatus](#)

### 8.598.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

#### Parameters

<i>injectTimeSyncStatus</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>
-----------------------------	---

### 8.598.2 Field Documentation

8.598.2.1 **ULONG** QmiCbkLocInjectTimeInd::injectTimeSyncStatus

## 8.599 QmiCbkLocInjectUTCTimeInd Struct Reference

### Data Fields

- [ULONG status](#)

### 8.599.1 Detailed Description

Contain the parameters passed for SetLocInjectUTCTimeCallback by the device.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Status of the UTC Time Injection 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> </ul> </li> </ul>
---------------	--

#### Note

None

### 8.599.2 Field Documentation

#### 8.599.2.1 ULONG QmiCbkLocInjectUTCTimeInd::status

## 8.600 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](#)

### 8.600.1 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>• Latitude (specified in WGS84 datum)</li> <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>• Longitude (specified in WGS84 datum)</li> <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 WGS84 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> <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> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </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

## 8.600.2 Field Documentation

- 8.600.2.1 **BYTE\*** QmiCbkLocPositionReportInd::pAltitudeAssumed
- 8.600.2.2 **ULONG\*** QmiCbkLocPositionReportInd::pAltitudeWrtEllipsoid
- 8.600.2.3 **ULONG\*** QmiCbkLocPositionReportInd::pAltitudeWrtMeanSeaLevel
- 8.600.2.4 **ULONG\*** QmiCbkLocPositionReportInd::pFixId
- 8.600.2.5 **gpsTime\*** QmiCbkLocPositionReportInd::pGpsTime
- 8.600.2.6 **ULONG\*** QmiCbkLocPositionReportInd::pHeading
- 8.600.2.7 **ULONG\*** QmiCbkLocPositionReportInd::pHeadingUnc
- 8.600.2.8 **BYTE\*** QmiCbkLocPositionReportInd::pHorConfidence
- 8.600.2.9 **ULONG\*** QmiCbkLocPositionReportInd::pHorReliability
- 8.600.2.10 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncCircular
- 8.600.2.11 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseOrientAzimuth
- 8.600.2.12 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMajor
- 8.600.2.13 **ULONG\*** QmiCbkLocPositionReportInd::pHorUncEllipseSemiMinor
- 8.600.2.14 **ULONGLONG\*** QmiCbkLocPositionReportInd::pLatitude
- 8.600.2.15 **BYTE\*** QmiCbkLocPositionReportInd::pLeapSeconds
- 8.600.2.16 **ULONGLONG\*** QmiCbkLocPositionReportInd::pLongitude
- 8.600.2.17 **ULONG\*** QmiCbkLocPositionReportInd::pMagneticDeviation
- 8.600.2.18 **precisionDilution\*** QmiCbkLocPositionReportInd::pPrecisionDilution
- 8.600.2.19 **sensorDataUsage\*** QmiCbkLocPositionReportInd::pSensorDataUsage
- 8.600.2.20 **ULONG\*** QmiCbkLocPositionReportInd::pSpeedHorizontal
- 8.600.2.21 **ULONG\*** QmiCbkLocPositionReportInd::pSpeedUnc
- 8.600.2.22 **ULONG\*** QmiCbkLocPositionReportInd::pSpeedVertical
- 8.600.2.23 **svUsedforFix\*** QmiCbkLocPositionReportInd::pSvUsedforFix
- 8.600.2.24 **ULONG\*** QmiCbkLocPositionReportInd::pTechnologyMask
- 8.600.2.25 **ULONG\*** QmiCbkLocPositionReportInd::pTimeSrc
- 8.600.2.26 **ULONGLONG\*** QmiCbkLocPositionReportInd::pTimestampUtc
- 8.600.2.27 **ULONG\*** QmiCbkLocPositionReportInd::pTimeUnc

8.600.2.28 **BYTE\*** QmiCbkLocPositionReportInd::pVertConfidence

8.600.2.29 **ULONG\*** QmiCbkLocPositionReportInd::pVertReliability

8.600.2.30 **ULONG\*** QmiCbkLocPositionReportInd::pVertUnc

8.600.2.31 **BYTE** QmiCbkLocPositionReportInd::sessionId

8.600.2.32 **ULONG** QmiCbkLocPositionReportInd::sessionStatus

## 8.601 QmiCbkLocSensorStreamingInd Struct Reference

### Data Fields

- [accelAcceptReady](#) \* [pAccelAcceptReady](#)
- [gyroAcceptReady](#) \* [pGyroAcceptReady](#)
- [accelTempAcceptReady](#) \* [pAccelTempAcceptReady](#)
- [gyroTempAcceptReady](#) \* [pGyroTempAcceptReady](#)

### 8.601.1 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>

### 8.601.2 Field Documentation

8.601.2.1 **accelAcceptReady\*** QmiCbkLocSensorStreamingInd::pAccelAcceptReady

8.601.2.2 **accelTempAcceptReady\*** QmiCbkLocSensorStreamingInd::pAccelTempAcceptReady

8.601.2.3 **gyroAcceptReady\*** QmiCbkLocSensorStreamingInd::pGyroAcceptReady

8.601.2.4 **gyroTempAcceptReady\*** QmiCbkLocSensorStreamingInd::pGyroTempAcceptReady

## 8.602 QmiCbkLocSetExtPowerConfigInd Struct Reference

### Data Fields

- [ULONG](#) status

### 8.602.1 Detailed Description

This structure contains LOC Set External Power Config Status

#### Parameters

<i>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>
---------------	---

### 8.602.2 Field Documentation

8.602.2.1 **ULONG** QmiCbKLocSetExtPowerConfigInd::status

## 8.603 QmiCbK NasLTECphyCalInfo Struct Reference

#### Data Fields

- [PhyCaAggScellIndType](#) sPhyCaAggScellIndType
- [PhyCaAggScellDIBw](#) sPhyCaAggScellDIBw
- [PhyCaAggScellInfo](#) sPhyCaAggScellInfo
- [PhyCaAggPcellInfo](#) sPhyCaAggPcellInfo
- [PhyCaAggScellIndex](#) sPhyCaAggScellIndex

### 8.603.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

#### Parameters

<i>pPhyCaAgg-ScellIndType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellIndType</a> for more information.</li> </ul>
<i>sPhyCaAgg-ScellDIBw</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellDIBw</a> for more information.</li> </ul>
<i>sPhyCaAgg-ScellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellInfo</a> for more information.</li> </ul>
<i>sPhyCaAgg-PcellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggPcellInfo</a> for more information.</li> </ul>

<i>sPhyCaAgg-ScellIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">PhyCaAggScellIndex</a> for more information.</li> </ul>
-----------------------------	--

### 8.603.2 Field Documentation

8.603.2.1 **PhyCaAggPcellInfo** QmiCbkNasLTECphyCalInfo::sPhyCaAggPcellInfo

8.603.2.2 **PhyCaAggScellIDIBw** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIDIBw

8.603.2.3 **PhyCaAggScellIndex** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIndex

8.603.2.4 **PhyCaAggScellIndType** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellIndType

8.603.2.5 **PhyCaAggScellInfo** QmiCbkNasLTECphyCalInfo::sPhyCaAggScellInfo

## 8.604 QmiCbkSwiOmaDmEventStatusReportInd Struct Reference

### Data Fields

- struct [sessionInfoTlv](#) SITlv

### 8.604.1 Field Documentation

8.604.1.1 struct **sessionInfoTlv** QmiCbkSwiOmaDmEventStatusReportInd::SITlv

## 8.605 QmiCbkSwiOmaDmEventStatusReportIndExt Struct Reference

### Data Fields

- struct [sessionInfoTlvExt](#) SITlv

### 8.605.1 Field Documentation

8.605.1.1 struct **sessionInfoTlvExt** QmiCbkSwiOmaDmEventStatusReportIndExt::SITlv

## 8.606 QmiCbkTmdMitiLvlRptInd Struct Reference

### Data Fields

- [\\_MitigationDevInfo](#) MitigationDevInfo
- [BYTE](#) currentMitigationLvl

### 8.606.1 Detailed Description

This structure contains LOC Cradle Mount Config Status

## Parameters

<i>MitigationDev-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">MitigationDevInfo</a> for more information.</li> </ul>
<i>current-MitigationLvl</i>	<ul style="list-style-type: none"> <li>• Current Thermal Mitigation Level</li> </ul>

## 8.606.2 Field Documentation

8.606.2.1 BYTE QmiCbkTmdMitiLvlRptInd::currentMitigationLvl

8.606.2.2 \_MitigationDevInfo QmiCbkTmdMitiLvlRptInd::MitigationDevInfo

## 8.607 QmiCbkWdsStatisticsIndState Struct Reference

## Data Fields

- [DataUlongTlv TxOkConutTlv](#)
- [DataUlongTlv RxOkConutTlv](#)
- [DataUlongLongTlv TxOkByteCountTlv](#)
- [DataUlongLongTlv RxOkByteCountTlv](#)
- [DataUlongTlv TxDropConutTlv](#)
- [DataUlongTlv RxDropConutTlv](#)

## 8.607.1 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>TxOkByteCount-Tlv</i>	<ul style="list-style-type: none"> <li>• Tx Ok Byte Count Packet Tlv Value.</li> </ul>
<i>RxOkByteCount-Tlv</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>

## 8.607.2 Field Documentation

8.607.2.1 **DataULongTlv** QmiCbkWdsStatisticsIndState::RxDropConutTlv

8.607.2.2 **DataULongLongTlv** QmiCbkWdsStatisticsIndState::RxOkByteCountTlv

8.607.2.3 **DataULongTlv** QmiCbkWdsStatisticsIndState::RxOkConutTlv

8.607.2.4 **DataULongTlv** QmiCbkWdsStatisticsIndState::TxDropConutTlv

8.607.2.5 **DataULongLongTlv** QmiCbkWdsStatisticsIndState::TxOkByteCountTlv

8.607.2.6 **DataULongTlv** QmiCbkWdsStatisticsIndState::TxOkConutTlv

## 8.608 qmifwinfo\_s Struct Reference

### Data Fields

- union {
  - struct [fwinfo\\_s](#) *g*
  - struct [slqsfwinfo\\_s](#) *s*
- } *dev*

### 8.608.1 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](#)

### 8.608.2 Field Documentation

8.608.2.1 union { ... } qmifwinfo\_s::dev

8.608.2.2 struct fwinfo\_s qmifwinfo\_s::g

8.608.2.3 struct slqsfwinfo\_s qmifwinfo\_s::s

## 8.609 QmiNas3GppNetworkInfo Struct Reference

### Data Fields

- [WORD pMCC](#)
- [WORD pMNC](#)
- [ULONG plnUse](#)
- [ULONG pRoaming](#)
- [ULONG pForbidden](#)
- [ULONG pPreferred](#)
- [CHAR pDescription](#) [255]

### 8.609.1 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>plnUse</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>

## 8.609.2 Field Documentation

8.609.2.1 **CHAR** QmiNas3GppNetworkInfo::pDescription[255]

8.609.2.2 **ULONG** QmiNas3GppNetworkInfo::pForbidden

8.609.2.3 **ULONG** QmiNas3GppNetworkInfo::pInUse

8.609.2.4 **WORD** QmiNas3GppNetworkInfo::pMCC

8.609.2.5 **WORD** QmiNas3GppNetworkInfo::pMNC

8.609.2.6 **ULONG** QmiNas3GppNetworkInfo::pPreferred

8.609.2.7 **ULONG** QmiNas3GppNetworkInfo::pRoaming

## 8.610 QmiNasGetRFBandInfoResp Struct Reference

### Data Fields

- struct qmTlvResult [results](#)
- **BYTE** \* [pInstancesSize](#)
- struct [RFBandInfoElements](#) \* [pRFBandInfoElements](#)

### 8.610.1 Field Documentation

8.610.1.1 **BYTE**\* QmiNasGetRFBandInfoResp::pInstancesSize

8.610.1.2 struct [RFBandInfoElements](#)\* QmiNasGetRFBandInfoResp::pRFBandInfoElements

8.610.1.3 struct qmTlvResult QmiNasGetRFBandInfoResp::results

## 8.611 QmiNasPerformNetworkScanResp Struct Reference

### Data Fields

- struct qmTlvResult [results](#)
- **BYTE** \* [pInstanceSize](#)
- struct [QmiNas3GppNetworkInfo](#) \* [pInstances](#)

### 8.611.1 Field Documentation

8.611.1.1 struct QmiNas3GppNetworkInfo\* QmiNasPerformNetworkScanResp::pInstances

8.611.1.2 BYTE\* QmiNasPerformNetworkScanResp::pInstanceSize

8.611.1.3 struct qmTlvResult QmiNasPerformNetworkScanResp::results

## 8.612 qmiSmsMessageList Struct Reference

### Data Fields

- uint32\_t [messageIndex](#)
- uint32\_t [messageTag](#)

### 8.612.1 Detailed Description

#### Parameters

<i>messageIndex</i>	<ul style="list-style-type: none"><li>• Message index of each matched message</li></ul>
<i>messageTag</i>	<ul style="list-style-type: none"><li>• Messagetag</li></ul>

### 8.612.2 Field Documentation

8.612.2.1 uint32\_t qmiSmsMessageList::messageIndex

8.612.2.2 uint32\_t qmiSmsMessageList::messageTag

## 8.613 qmiWSDDataBearerTechnology Struct Reference

### Data Fields

- uint8\_t [currentNetwork](#)
- uint32\_t [ratMask](#)
- uint32\_t [soMask](#)

### 8.613.1 Detailed Description

#### Parameters

<i>currentNetwork</i>	current selected network
<i>Radio</i>	Access Technology (RAT) mask
<i>soMask</i>	Service Option (SO) mask

### 8.613.2 Field Documentation

8.613.2.1 uint8\_t qmiWSDDataBearerTechnology::currentNetwork

8.613.2.2 uint32\_t qmiWSDDataBearerTechnology::ratMask

8.613.2.3 uint32\_t qmiWSDDataBearerTechnology::soMask

## 8.614 QmiWdsIpAddressInfo Struct Reference

### Data Fields

- [ULONG](#) \* [pIPAddressV4](#)
- [USHORT](#) \* [pIPAddressV6](#)
- [BYTE](#) \* [pIPv6prefixlen](#)

### 8.614.1 Detailed Description

#### Parameters

<i>pIPAddressV4</i> [- OUT]	<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> [- OUT]	<ul style="list-style-type: none"> <li>• Current IPv6 address Space for storing the 8 element array of type USHORT for the IPv6 address is allocated by the application. The IP Address is stored in the user supplied buffer as follows: User buffer: [&lt;U0&gt;..&lt;&lt;U7&gt;] IPv6 address from the network: 1234:2A01:.....:5678 User buffer contents: U0 corresponds to 1234 U1 corresponds to 2A01 ----- ----- U7 corresponds to 5678</li> </ul>
<i>pIPv6prefixlen</i> [- OUT]	<ul style="list-style-type: none"> <li>• IPv6 prefix length in number of bits</li> </ul>

### 8.614.2 Field Documentation

8.614.2.1 **ULONG\*** QmiWdsIpAddressInfo::pIPAddressV4

8.614.2.2 **USHORT\*** QmiWdsIpAddressInfo::pIPAddressV6

8.614.2.3 **BYTE\*** QmiWdsIpAddressInfo::pIPv6prefixlen

## 8.615 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](#)

### 8.615.1 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 One or more bytes describing the profile. Description may be a user-defined name for the profile. QMI_ERR_ARG_TOO_LONG is returned if profile_name is too long.</li> </ul>
<i>pPDPTyep</i>	<ul style="list-style-type: none"> <li>• PDP type <ul style="list-style-type: none"> <li>– 0 – PDP-IP (IPv4)</li> <li>– 1 - PDP-PPP</li> <li>– 2 - PDP-IPv6</li> <li>– 3 - PDP-IPv4v6</li> <li>– 0xffffffff - invalid</li> </ul> </li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name String parameter that is a logical name used to select the GGSN and external packet data network. If value is NULL or omitted, then the subscription default value is requested. QMI_ERR_ARG_TOO_LONG is returned if the APN name is too long.</li> </ul>
<i>pPrimaryDNSV4</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address</li> </ul>

<i>pSecondaryDN-SV4</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address</li> </ul>
<i>pUMTSGranted-QoS</i>	<ul style="list-style-type: none"> <li>• UMTS Granted QoS</li> </ul>
<i>pGPRSGranted-QoS</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>pPCSCFFQDN-AddrList</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>

## 8.615.2 Field Documentation

8.615.2.1 **CHAR\*** qmiWdsRunTimeSettings::pAPNName

8.615.2.2 **ULONG\*** qmiWdsRunTimeSettings::pAuthentication

8.615.2.3 **struct DomainNameList\*** qmiWdsRunTimeSettings::pDomainList

8.615.2.4 **struct GPRSQoS\*** qmiWdsRunTimeSettings::pGPRSGrantedQoS

8.615.2.5 **ULONG\*** qmiWdsRunTimeSettings::pGWAddressV4

8.615.2.6 **BYTE\*** qmiWdsRunTimeSettings::pIMCNflag

8.615.2.7 **ULONG\*** qmiWdsRunTimeSettings::pIPAddressV4

8.615.2.8 **BYTE\*** qmiWdsRunTimeSettings::pIPFamilyPreference

8.615.2.9 **struct IPV6AddressInfo\*** qmiWdsRunTimeSettings::pIPV6AddrInfo

8.615.2.10 **struct IPV6GWAddressInfo\*** qmiWdsRunTimeSettings::pIPV6GWAddrInfo

8.615.2.11 **ULONG\*** qmiWdsRunTimeSettings::pMtu

8.615.2.12 **BYTE\*** qmiWdsRunTimeSettings::pPCSCFAddrPCO

8.615.2.13 **struct PCSCFFQDNAddressList\*** qmiWdsRunTimeSettings::pPCSCFFQDNAddrList

- 8.615.2.14 **ULONG\*** qmiWdsRunTimeSettings::pPDPTType
- 8.615.2.15 **ULONG\*** qmiWdsRunTimeSettings::pPrimaryDNSV4
- 8.615.2.16 **USHORT\*** qmiWdsRunTimeSettings::pPrimaryDNSV6
- 8.615.2.17 **struct ProfileIdentifier\*** qmiWdsRunTimeSettings::pProfileID
- 8.615.2.18 **CHAR\*** qmiWdsRunTimeSettings::pProfileName
- 8.615.2.19 **ULONG\*** qmiWdsRunTimeSettings::pSecondaryDNSV4
- 8.615.2.20 **USHORT\*** qmiWdsRunTimeSettings::pSecondaryDNSV6
- 8.615.2.21 **struct PCSCFIPv4ServerAddressList\*** qmiWdsRunTimeSettings::pServerAddrList
- 8.615.2.22 **ULONG\*** qmiWdsRunTimeSettings::pSubnetMaskV4
- 8.615.2.23 **WORD\*** qmiWdsRunTimeSettings::pTechnology
- 8.615.2.24 **struct UMTSQoS\*** qmiWdsRunTimeSettings::pUMTSGrantedQoS
- 8.615.2.25 **CHAR\*** qmiWdsRunTimeSettings::pUsername

## 8.616 QosClassID Struct Reference

### Data Fields

- [BYTE QCI](#)
- [ULONG gDIBitRate](#)
- [ULONG maxDIBitRate](#)
- [ULONG gUIBitRate](#)
- [ULONG maxUIBitRate](#)

### 8.616.1 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>gDIBitRate</i>	<ul style="list-style-type: none"> <li>• Guaranteed DL bit rate</li> </ul>
<i>maxDIBitRate</i>	<ul style="list-style-type: none"> <li>• maxDIBitRate</li> </ul>

<i>gUIBitRate</i>	<ul style="list-style-type: none"><li>• Guaranteed UL bit rate</li></ul>
<i>maxUIBitRate</i>	<ul style="list-style-type: none"><li>• Maximum UL bit rate</li></ul>

## 8.616.2 Field Documentation

8.616.2.1 **ULONG** QosClassID::gDIBitRate

8.616.2.2 **ULONG** QosClassID::gUIBitRate

8.616.2.3 **ULONG** QosClassID::maxDIBitRate

8.616.2.4 **ULONG** QosClassID::maxUIBitRate

8.616.2.5 **BYTE** QosClassID::QCI

## 8.617 QosEventInfo Struct Reference

### Data Fields

- **ULONG** \* [pDataBearer](#)
- **ULONG** \* [pPacketsCountTX](#)
- **ULONG** \* [pPacketsCountRX](#)
- **ULONGLONG** \* [pTotalBytesTX](#)
- **ULONGLONG** \* [pTotalBytesRX](#)

### 8.617.1 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>

## 8.617.2 Field Documentation

8.617.2.1 **ULONG\*** QosEventInfo::pDataBearer8.617.2.2 **ULONG\*** QosEventInfo::pPacketsCountRX8.617.2.3 **ULONG\*** QosEventInfo::pPacketsCountTX8.617.2.4 **ULONGLONG\*** QosEventInfo::pTotalBytesRX

8.617.2.5 **ULONGLONG\*** QosEventInfo::pTotalBytesTX

## 8.618 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

### 8.618.1 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>

### 8.618.2 Field Documentation

8.618.2.1 **BYTE\*** QosFlowInfo::pBearerID

8.618.2.2 **QosFlowInfoState\*** QosFlowInfo::pQFlowState

8.618.2.3 **swiQosFilter\*** QosFlowInfo::pRxQFilter[MAX\_QOS\_FILTER\_TLV]

8.618.2.4 **swiQosFlow\*** QosFlowInfo::pRxQFlowGranted

8.618.2.5 **swiQosFilter\*** QosFlowInfo::pTxQFilter[MAX\_QOS\_FILTER\_TLV]

8.618.2.6 `swiQosFlow*` `QosFlowInfo::pTxQFlowGranted`

## 8.619 QosFlowInfoState Struct Reference

### Data Fields

- [ULONG id](#)
- [BYTE isNewFlow](#)
- [BYTE state](#)

### 8.619.1 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>	This indicates that the flow that was added/modified/deleted: <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>

### 8.619.2 Field Documentation

8.619.2.1 **ULONG** `QosFlowInfoState::id`

8.619.2.2 **BYTE** `QosFlowInfoState::isNewFlow`

8.619.2.3 **BYTE** `QosFlowInfoState::state`

## 8.620 QosMap Struct Reference

### Data Fields

- [BYTE dscp](#)
- [ULONG qos\\_id](#)
- [BYTE state](#)

### 8.620.1 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>

## 8.620.2 Field Documentation

8.620.2.1 BYTE QosMap::dscp

8.620.2.2 ULONG QosMap::qos\_id

8.620.2.3 BYTE QosMap::state

## 8.621 RankIndicatorInd Struct Reference

## Data Fields

- [WORD Count1](#)
- [WORD Count2](#)

### 8.621.1 Field Documentation

8.621.1.1 WORD RankIndicatorInd::Count1

8.621.1.2 WORD RankIndicatorInd::Count2

## 8.622 readResult Struct Reference

## Data Fields

- [WORD contentLen](#)
- [BYTE content](#) [255+1]

### 8.622.1 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>

### 8.622.2 Field Documentation

8.622.2.1 **BYTE** readResult::content[255+1]

8.622.2.2 **WORD** readResult::contentLen

## 8.623 readTransparentInfo Struct Reference

### Data Fields

- [WORD offset](#)
- [WORD length](#)

### 8.623.1 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>

### 8.623.2 Field Documentation

8.623.2.1 **WORD** readTransparentInfo::length

8.623.2.2 **WORD** readTransparentInfo::offset

## 8.624 redirNumInfo Struct Reference

### Data Fields

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

### 8.624.1 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 follow.</li> </ul>
<i>number[255]</i>	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.624.2 Field Documentation

8.624.2.1 **BYTE** `redirNumInfo::number[255]`

8.624.2.2 **BYTE** `redirNumInfo::numLen`

8.624.2.3 **BYTE** `redirNumInfo::numPlan`

8.624.2.4 **BYTE** `redirNumInfo::numType`

8.624.2.5 **BYTE** `redirNumInfo::PI`

8.624.2.6 **BYTE** `redirNumInfo::reason`

8.624.2.7 **BYTE** `redirNumInfo::SI`

## 8.625 registerRefresh Struct Reference

### Data Fields

- [BYTE](#) `registerFlag`
- [BYTE](#) `voteForInit`
- [WORD](#) `numFiles`
- [fileInfo](#) `arrfileInfo` [255]

### 8.625.1 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>– <code>file_id</code></li> <li>– <code>path_len</code></li> <li>– <code>path</code></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>

### 8.625.2 Field Documentation

8.625.2.1 `fileInfo` registerRefresh::arrfileInfo[255]

8.625.2.2 `WORD` registerRefresh::numFiles

8.625.2.3 `BYTE` registerRefresh::registerFlag

8.625.2.4 `BYTE` registerRefresh::voteForInit

## 8.626 remainingRetries Struct Reference

### Data Fields

- [BYTE](#) `verifyLeft`
- [BYTE](#) `unlockLeft`

### 8.626.1 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>unlockLeft</i>	<ul style="list-style-type: none"><li>• Number of remaining attempts to unlock 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.

### 8.626.2 Field Documentation

8.626.2.1 `BYTE` remainingRetries::unlockLeft

8.626.2.2 `BYTE` remainingRetries::verifyLeft

## 8.627 remotePartyName Struct Reference

### Data Fields

- [BYTE](#) `namePI`
- [BYTE](#) `codingScheme`
- [BYTE](#) `nameLen`
- [BYTE](#) `callerName` [255]

### 8.627.1 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>

### 8.627.2 Field Documentation

8.627.2.1 **BYTE** remotePartyName::callerName[255]

8.627.2.2 **BYTE** remotePartyName::codingScheme

8.627.2.3 **BYTE** remotePartyName::nameLen

8.627.2.4 **BYTE** remotePartyName::namePI

## 8.628 remotePartyNum Struct Reference

### Data Fields

- [BYTE presentationInd](#)
- [BYTE numLen](#)
- [BYTE remPartyNumber](#) [81]

### 8.628.1 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>

## 8.628.2 Field Documentation

8.628.2.1 BYTE remotePartyNum::numLen

8.628.2.2 BYTE remotePartyNum::presentationInd

8.628.2.3 BYTE remotePartyNum::remPartyNumber[81]

## 8.629 ReqFieldsList Struct Reference

## Data Fields

- [BYTE requestFieldsLen](#)
- [BYTE requestFields \[256\]](#)

## 8.629.1 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>

## 8.629.2 Field Documentation

8.629.2.1 BYTE ReqFieldsList::requestFields[256]

8.629.2.2 BYTE ReqFieldsList::requestFieldsLen

## 8.630 RespFieldsList Struct Reference

### Data Fields

- [BYTE responseFieldsLen](#)
- [BYTE responseFields](#) [256]

### 8.630.1 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>

### 8.630.2 Field Documentation

8.630.2.1 BYTE RespFieldsList::responseFields[256]

8.630.2.2 BYTE RespFieldsList::responseFieldsLen

## 8.631 RFBandInfoElements Struct Reference

### Data Fields

- [BYTE radioInterface](#)
- [WORD activeBandClass](#)
- [WORD activeChannel](#)
- [uint8\\_t radioInterface](#)
- [uint16\\_t activeBandClass](#)
- [uint16\\_t activeChannel](#)

### 8.631.1 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>
<i>radiolInterface</i>	radio interface technology
<i>activeBandClass</i>	active band class
<i>activeChannel</i>	active channel

### 8.631.2 Field Documentation

8.631.2.1 WORD RFBandInfoElements::activeBandClass

8.631.2.2 uint16\_t RFBandInfoElements::activeBandClass

8.631.2.3 WORD RFBandInfoElements::activeChannel

8.631.2.4 uint16\_t RFBandInfoElements::activeChannel

8.631.2.5 BYTE RFBandInfoElements::radiolInterface

8.631.2.6 uint8\_t RFBandInfoElements::radiolInterface

## 8.632 rmTrasnferStaticsReq Struct Reference

### Data Fields

- uint8\_t [bResetStatistics](#)
- uint32\_t [ulMask](#)

### 8.632.1 Detailed Description

#### Parameters

<i>bResetStatistics</i>	Clear RM statistics
<i>ulMask</i>	Requested statistic bit mask

### 8.632.2 Field Documentation

8.632.2.1 uint8\_t rmTrasnferStaticsReq::bResetStatistics

8.632.2.2 uint32\_t rmTrasnferStaticsReq::ulMask

## 8.633 roamIndList Struct Reference

### Data Fields

- BYTE [numInstances](#)
- BYTE [radiolInterface](#) [0x0A]
- BYTE [roamIndicator](#) [0x0A]

### 8.633.1 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>radioInterface</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>

### 8.633.2 Field Documentation

8.633.2.1 **BYTE** roamIndList::numInstances

8.633.2.2 **BYTE** roamIndList::radioInterface[0x0A]

8.633.2.3 **BYTE** roamIndList::roamIndicator[0x0A]

## 8.634 RoamingInfo Struct Reference

#### Data Fields

- [BYTE TlvPresent](#)
- [BYTE roaming\\_ind](#)

### 8.634.1 Field Documentation

8.634.1.1 **BYTE** RoamingInfo::roaming\_ind

8.634.1.2 **BYTE** RoamingInfo::TlvPresent

## 8.635 roamTimer Struct Reference

## Data Fields

- [BYTE namID](#)
- [ULONG roamTimerValue](#)

### 8.635.1 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>

### 8.635.2 Field Documentation

8.635.2.1 **BYTE** roamTimer::namID

8.635.2.2 **ULONG** roamTimer::roamTimerValue

## 8.636 RSRPThresh Struct Reference

## Data Fields

- [BYTE RSRPThresListLen](#)
- [SHORT \\* pRSRPThresList](#)

### 8.636.1 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>

## 8.636.2 Field Documentation

8.636.2.1 **SHORT\*** RSRPThresh::pRSRPThresList8.636.2.2 **BYTE** RSRPThresh::RSRPThresListLen

## 8.637 rsrqInformation Struct Reference

## Data Fields

- [INT8 rsrq](#)
- [BYTE radiolf](#)

## 8.637.1 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>

## 8.637.2 Field Documentation

8.637.2.1 **BYTE** rsrqInformation::radiolf8.637.2.2 **INT8** rsrqInformation::rsrq

## 8.638 RSRQThresh Struct Reference

## Data Fields

- [BYTE RSRQThresListLen](#)
- [INT8 \\* pRSRQThresList](#)

### 8.638.1 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>

### 8.638.2 Field Documentation

8.638.2.1 [INT8\\*](#) RSRQThresh::pRSRQThresList

8.638.2.2 [BYTE](#) RSRQThresh::RSRQThresListLen

## 8.639 RSSIThresh Struct Reference

## Data Fields

- [BYTE RSSIThresListLen](#)
- [INT8 \\* pRSSIThresList](#)

### 8.639.1 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>

## 8.639.2 Field Documentation

8.639.2.1 INT8\* RSSIThresh::pRSSIThresList

8.639.2.2 BYTE RSSIThresh::RSSIThresListLen

## 8.640 RXAGCList Struct Reference

## Data Fields

- WORD \* [pRXStaticGain](#)
- WORD \* [pRXAIG](#)
- WORD \* [pRXExpThres](#)
- WORD \* [pRXExpSlope](#)
- WORD \* [pRXComprThres](#)
- WORD \* [pRXComprSlope](#)

## 8.640.1 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>

## 8.640.2 Field Documentation

8.640.2.1 **WORD\*** RXAGCList::pRXAIG

8.640.2.2 **WORD\*** RXAGCList::pRXComprSlope

8.640.2.3 **WORD\*** RXAGCList::pRXComprThres

8.640.2.4 **WORD\*** RXAGCList::pRXExpSlope

8.640.2.5 **WORD\*** RXAGCList::pRXExpThres

8.640.2.6 **WORD\*** RXAGCList::pRXStaticGain

## 8.641 RXAVCList Struct Reference

### Data Fields

- **WORD \*** [pAVRXAVCSens](#)
- **WORD \*** [pAVRXAVCHeadroom](#)

### 8.641.1 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>

## 8.641.2 Field Documentation

8.641.2.1 **WORD\*** RXAVCList::pAVRXAVCHeadroom

8.641.2.2 **WORD\*** RXAVCList::pAVRXAVCSens

## 8.642 rxInfo Struct Reference

### Data Fields

- **BYTE** [isRadioTuned](#)

- [INT32 rxPower](#)
- [INT32 ecio](#)
- [INT32 rscp](#)
- [INT32 rsrp](#)
- [ULONG phase](#)

### 8.642.1 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.</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.</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.</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>• Current reference signal received power in 1/10 dbm valid for LTE.</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>

### 8.642.2 Field Documentation

8.642.2.1 INT32 rxInfo::ecio

8.642.2.2 BYTE rxInfo::isRadioTuned

8.642.2.3 ULONG rxInfo::phase

8.642.2.4 INT32 rxInfo::rscp

8.642.2.5 INT32 rxInfo::rsrp

8.642.2.6 INT32 rxInfo::rxPower

## 8.643 RXPCMIIRFitr Struct Reference

### Data Fields

- WORD \* pFlag
- WORD \* pStageCnt
- BYTE \* pStage0Val
- BYTE \* pStage1Val
- BYTE \* pStage2Val
- BYTE \* pStage3Val
- BYTE \* pStage4Val

### 8.643.1 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>

### 8.643.2 Field Documentation

8.643.2.1 WORD\* RXPCMIIRFitr::pFlag

8.643.2.2 BYTE\* RXPCMIIRFitr::pStage0Val

8.643.2.3 BYTE\* RXPCMIIRFitr::pStage1Val

8.643.2.4 BYTE\* RXPCMIIRFitr::pStage2Val

8.643.2.5 BYTE\* RXPCMIIRFitr::pStage3Val

8.643.2.6 BYTE\* RXPCMIIRFitr::pStage4Val

8.643.2.7 WORD\* RXPCMIIRFitr::pStageCnt

## 8.644 rxSignalStrengthListElement Struct Reference

### Data Fields

- [SHORT rxSignalStrength](#)
- [BYTE radiolf](#)

### 8.644.1 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.

**8.644.2 Field Documentation**

8.644.2.1 **BYTE** rxSignalStrengthListElement::radioIf

8.644.2.2 **SHORT** rxSignalStrengthListElement::rxSignalStrength

**8.645 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

**8.645.1 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>

### 8.645.2 Field Documentation

8.645.2.1 **BYTE** sApnExtraParams::ambr\_dl

8.645.2.2 **BYTE** sApnExtraParams::ambr\_dl\_ext

8.645.2.3 **BYTE** sApnExtraParams::ambr\_dl\_ext2

8.645.2.4 **BYTE** sApnExtraParams::ambr\_ul

8.645.2.5 **BYTE** sApnExtraParams::ambr\_ul\_ext

8.645.2.6 **BYTE** sApnExtraParams::ambr\_ul\_ext2

8.645.2.7 **ULONG** sApnExtraParams::apnId

## 8.646 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

### 8.646.1 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

## 8.646.2 Field Documentation

8.646.2.1 FLOAT satelliteInfo::azimuth

8.646.2.2 FLOAT satelliteInfo::elevation

8.646.2.3 WORD satelliteInfo::gnssSvId

8.646.2.4 BYTE satelliteInfo::healthStatus

8.646.2.5 FLOAT satelliteInfo::snr

8.646.2.6 BYTE satelliteInfo::svInfoMask

8.646.2.7 BYTE satelliteInfo::svListLen

8.646.2.8 ULONG satelliteInfo::svStatus

8.646.2.9 ULONG satelliteInfo::system

8.646.2.10 ULONG satelliteInfo::validMask

## 8.647 sensorData Struct Reference

## Data Fields

- ULONG timeOfFirstSample
- BYTE flags
- BYTE sensorDataLen

- [WORD timeOffset](#) [64]
- [ULONG xAxis](#) [64]
- [ULONG yAxis](#) [64]
- [ULONG zAxis](#) [64]

### 8.647.1 Detailed Description

This structure specifies information regarding the 3-Axis Sensor Data.

#### Parameters

<i>timeOfFirst-Sample</i>	<ul style="list-style-type: none"> <li>• Denotes a full 32-bit time stamp of the first (oldest) sample in this message.</li> <li>• The time stamp is in the time reference scale that is used by the sensor time source.</li> <li>• Units - Milliseconds</li> </ul>
<i>flags</i>	<ul style="list-style-type: none"> <li>• Flags to indicate any deviation from the default measurement assumptions.</li> <li>• All unused bits in this field must be set to 0.</li> <li>• Valid bitmasks <ul style="list-style-type: none"> <li>– 0x01 - Bitmask to specify that a sign reversal is required while interpreting the sensor data; only applies to the accelerometer samples</li> <li>– 0x02 - Bitmask to specify that the sensor time stamp is the same as the modem time stamp</li> </ul> </li> </ul>
<i>sensorDataLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– timeOffset</li> <li>– xAxis</li> <li>– yAxis</li> <li>– zAxis</li> </ul> </li> </ul>
<i>timeOffset</i>	<ul style="list-style-type: none"> <li>• Sample time offset</li> <li>• Units - Milliseconds</li> </ul>
<i>xAxis</i>	<ul style="list-style-type: none"> <li>• Sensor x-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>
<i>yAxis</i>	<ul style="list-style-type: none"> <li>• Sensor Y-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>
<i>zAxis</i>	<ul style="list-style-type: none"> <li>• Sensor Z-axis sample.</li> <li>• Units Accelerometer - Meters/seconds square</li> <li>• Units Gyroscope - Radians/second</li> </ul>

### 8.647.2 Field Documentation

8.647.2.1 **BYTE** sensorData::flags

8.647.2.2 **BYTE** sensorData::sensorDataLen

8.647.2.3 **ULONG** sensorData::timeOfFirstSample

8.647.2.4 **WORD** sensorData::timeOffset[64]

8.647.2.5 **ULONG** sensorData::xAxis[64]

8.647.2.6 **ULONG** sensorData::yAxis[64]

8.647.2.7 **ULONG** sensorData::zAxis[64]

## 8.648 sensorDataUsage\_s Struct Reference

### Data Fields

- [ULONG](#) `usageMask`
- [ULONG](#) `aidingIndicatorMask`

### 8.648.1 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

### 8.648.2 Field Documentation

8.648.2.1 **ULONG** sensorDataUsage\_s::aidingIndicatorMask

8.648.2.2 **ULONG** sensorDataUsage\_s::usageMask

## 8.649 serialNumbersInfo Struct Reference

### Data Fields

- [BYTE esnSize](#)
- [CHAR \\* pESNString](#)
- [BYTE imeiSize](#)
- [CHAR \\* pIMEIString](#)
- [BYTE meidSize](#)
- [CHAR \\* pMEIDString](#)
- [BYTE imeiSvnSize](#)
- [CHAR \\* plmeiSvnString](#)

### 8.649.1 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>plmeiSvnString[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>

### 8.649.2 Field Documentation

#### 8.649.2.1 BYTE serialNumbersInfo::esnSize

8.649.2.2 **BYTE** serialNumbersInfo::imeiSize

8.649.2.3 **BYTE** serialNumbersInfo::imeiSvnSize

8.649.2.4 **BYTE** serialNumbersInfo::meidSize

8.649.2.5 **CHAR\*** serialNumbersInfo::pESNString

8.649.2.6 **CHAR\*** serialNumbersInfo::pIMEIString

8.649.2.7 **CHAR\*** serialNumbersInfo::pImeiSvnString

8.649.2.8 **CHAR\*** serialNumbersInfo::pMEIDString

## 8.650 serviceProviderName Struct Reference

### Data Fields

- [BYTE displayCondition](#)
- [BYTE spnLength](#)
- [BYTE spn \[255\]](#)

### 8.650.1 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>

### 8.650.2 Field Documentation

8.650.2.1 **BYTE** serviceProviderName::displayCondition

8.650.2.2 **BYTE** serviceProviderName::spn[255]

8.650.2.3 **BYTE** serviceProviderName::spnLength

## 8.651 ServingSystemInfo Struct Reference

### Data Fields

- [BYTE registrationState](#)
- [BYTE csAttachState](#)

- [BYTE psAttachState](#)
- [BYTE selectedNetwork](#)
- [BYTE radiolInterfaceNo](#)
- [BYTE radiolInterfaceList](#) [255]
- [BYTE hdrPersonality](#)

### 8.651.1 Detailed Description

This structure will hold the serving system parameters information

#### Parameters

<i>registrationState</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>csAttachState</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>psAttachState</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>selectedNetwork</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>radiolInterfaceNo</i>	<ul style="list-style-type: none"> <li>- Number of radio interfaces currently in use; this indicates how many radio_if identifiers follow this field</li> </ul>
<i>radiolInterface-List</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>
<i>hdrPersonality</i>	<ul style="list-style-type: none"> <li>- 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> </li> </ul>

Note: None

## 8.651.2 Field Documentation

8.651.2.1 **BYTE** ServingSystemInfo::csAttachState

8.651.2.2 **BYTE** ServingSystemInfo::hdrPersonality

8.651.2.3 **BYTE** ServingSystemInfo::psAttachState

8.651.2.4 **BYTE** ServingSystemInfo::radiolInterfaceList[255]

8.651.2.5 **BYTE** ServingSystemInfo::radiolInterfaceNo

8.651.2.6 **BYTE** ServingSystemInfo::registrationState

8.651.2.7 **BYTE** ServingSystemInfo::selectedNetwork

## 8.652 servSystem Struct Reference

### Data Fields

- [BYTE](#) regState
- [BYTE](#) csAttachState
- [BYTE](#) psAttachState
- [BYTE](#) selNetwork
- [BYTE](#) numRadiolInterfaces
- [BYTE](#) radiolInterface [0x0A]

### 8.652.1 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_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>

## 8.652.2 Field Documentation

8.652.2.1 **BYTE** servSystem::csAttachState

8.652.2.2 **BYTE** servSystem::numRadioInterfaces

8.652.2.3 **BYTE** servSystem::psAttachState

8.652.2.4 **BYTE** servSystem::radioInterface[0x0A]

8.652.2.5 **BYTE** servSystem::regState

8.652.2.6 **BYTE** servSystem::selNetwork

## 8.653 sessionInfo Union Reference

### Data Fields

- struct [omaDmFotaTlv](#) [omaDmFota](#)
- struct [omaDmConfigTlv](#) [omaDmConfig](#)

- struct [omaDmNotificationsTlv](#) [omaDmNotifications](#)

### 8.653.1 Detailed Description

This union [sessionInfo](#) consist of [omaDmFotaTlv](#), [omaDmConfigTlv](#) and [omaDmNotificationsTlv](#), out of which one will be unpacked against pEventFields.

### 8.653.2 Field Documentation

8.653.2.1 struct [omaDmConfigTlv](#) [sessionInfo::omaDmConfig](#)

8.653.2.2 struct [omaDmFotaTlv](#) [sessionInfo::omaDmFota](#)

8.653.2.3 struct [omaDmNotificationsTlv](#) [sessionInfo::omaDmNotifications](#)

## 8.654 sessionInfoExt Union Reference

### Data Fields

- struct [omaDmFotaTlvExt](#) [omaDmFota](#)
- struct [omaDmConfigTlvExt](#) [omaDmConfig](#)

### 8.654.1 Detailed Description

This union [sessionInfo](#) consist of [omaDmFotaTlv](#) and [omaDmConfigTlv](#), out of which one will be unpacked against pEventFields.

### 8.654.2 Field Documentation

8.654.2.1 struct [omaDmConfigTlvExt](#) [sessionInfoExt::omaDmConfig](#)

8.654.2.2 struct [omaDmFotaTlvExt](#) [sessionInfoExt::omaDmFota](#)

## 8.655 sessionInfoTlv Struct Reference

### Data Fields

- [BYTE](#) [TlvPresent](#)
- [ULONG](#) [sessionType](#)
- [sessionInformation](#) [sessionInfo](#)

### 8.655.1 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

## 8.655.2 Field Documentation

8.655.2.1 **sessionInformation** sessionInfoTlv::sessionInfo

8.655.2.2 **ULONG** sessionInfoTlv::sessionType

8.655.2.3 **BYTE** sessionInfoTlv::TlvPresent

## 8.656 sessionInfoTlvExt Struct Reference

### Data Fields

- [BYTE](#) TlvPresent
- [ULONG](#) sessionType
- [sessionInformationExt](#) sessionInfo

### 8.656.1 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

## 8.656.2 Field Documentation

8.656.2.1 **sessionInformationExt** sessionInfoTlvExt::sessionInfo

8.656.2.2 **ULONG** sessionInfoTlvExt::sessionType

8.656.2.3 **BYTE** sessionInfoTlvExt::TlvPresent

## 8.657 SetAudioPathConfigReq Struct Reference

### Data Fields

- [BYTE](#) Profile
- [BYTE](#) \* pECMode
- [BYTE](#) \* pNSEnable
- [WORD](#) \* pTXGain
- [WORD](#) \* pDTMFTXGain
- [WORD](#) \* pCodecSTGain
- [TXPCMIIRFiltr](#) \* pTXPCMIIRFiltr
- [RXPCMIIRFiltr](#) \* pRXPCMIIRFiltr
- [BYTE](#) \* pRXAVCAGCSwitch
- [BYTE](#) \* pTXAVCSwitch
- [RXAGCList](#) \* pRXAGCList
- [RXAVCList](#) \* pRXAVCList
- [TXAGCList](#) \* pTXAGCList

### 8.657.1 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_CODECASTG <ul style="list-style-type: none"> <li>0x0000 - 0xffff</li> </ul> </li> </ul>
<i>pTXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">TXPCMIIRFiltr</a> for more information</li> </ul>
<i>pRXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">RXPCMIIRFiltr</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>
-------------------	---

## 8.657.2 Field Documentation

8.657.2.1 **WORD\*** SetAudioPathConfigReq::pCodecSTGain

8.657.2.2 **WORD\*** SetAudioPathConfigReq::pDTMFTXGain

8.657.2.3 **BYTE\*** SetAudioPathConfigReq::pECMode

8.657.2.4 **BYTE\*** SetAudioPathConfigReq::pNSEnable

8.657.2.5 **BYTE** SetAudioPathConfigReq::Profile

8.657.2.6 **RXAGCList\*** SetAudioPathConfigReq::pRXAGCList

8.657.2.7 **BYTE\*** SetAudioPathConfigReq::pRXAVCAGCSwitch

8.657.2.8 **RXAVCList\*** SetAudioPathConfigReq::pRXAVCList

8.657.2.9 **RXPCMIIRFitr\*** SetAudioPathConfigReq::pRXPCMIIRFitr

8.657.2.10 **TXAGCList\*** SetAudioPathConfigReq::pTXAGCList

8.657.2.11 **BYTE\*** SetAudioPathConfigReq::pTXAVCSwitch

8.657.2.12 **WORD\*** SetAudioPathConfigReq::pTXGain

8.657.2.13 **TXPCMIIRFitr\*** SetAudioPathConfigReq::pTXPCMIIRFitr

## 8.658 SetAudioProfileReq Struct Reference

### Data Fields

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

### 8.658.1 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>

## 8.658.2 Field Documentation

8.658.2.1 BYTE SetAudioProfileReq::EarMute

8.658.2.2 BYTE SetAudioProfileReq::Generator

8.658.2.3 BYTE SetAudioProfileReq::MicMute

8.658.2.4 BYTE SetAudioProfileReq::Profile

8.658.2.5 BYTE SetAudioProfileReq::Volume

## 8.659 SetAudioVoITLBConfigReq Struct Reference

## Data Fields

- [BYTE Profile](#)

- [BYTE Generator](#)
- [BYTE Volume](#)
- [BYTE Item](#)
- [WORD VolValue](#)

### 8.659.1 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>

### 8.659.2 Field Documentation

8.659.2.1 **BYTE** SetAudioVolTLBConfigReq::Generator

8.659.2.2 **BYTE** SetAudioVolTLBConfigReq::Item

8.659.2.3 **BYTE** SetAudioVolTLBConfigReq::Profile

8.659.2.4 **BYTE** SetAudioVolTLBConfigReq::Volume

8.659.2.5 **WORD** SetAudioVolTLBConfigReq::VolValue

## 8.660 SetAudioVolTLBConfigResp Struct Reference

#### Data Fields

- [WORD ResCode](#)

### 8.660.1 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>
----------------	--

### 8.660.2 Field Documentation

#### 8.660.2.1 WORD SetAudioVolTLBConfigResp::ResCode

## 8.661 setCustomSettingV2 Struct Reference

#### Data Fields

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

### 8.661.1 Detailed Description

This structure contains customization settings set to modem

#### Parameters

<i>cust_id</i> [IN]	<ul style="list-style-type: none"> <li>Customization ID (Maximum 64 bytes)</li> </ul>
<i>value_length</i> [IN]	<ul style="list-style-type: none"> <li>length of <a href="#">cust_value</a> field</li> </ul>
<i>cust_value</i> [IN]	<ul style="list-style-type: none"> <li>Customization Setting Value (Maximum 8 bytes)</li> </ul>

### 8.661.2 Field Documentation

#### 8.661.2.1 CHAR setCustomSettingV2::cust\_id[64+1]

#### 8.661.2.2 BYTE setCustomSettingV2::cust\_value[8+1]

#### 8.661.2.3 WORD setCustomSettingV2::value\_length

## 8.662 setDyingGaspCfg Struct Reference

#### Data Fields

- BYTE \* [pDestSMSNum](#)
- BYTE \* [pDestSMSContent](#)

### 8.662.1 Detailed Description

This struture contains the TLV required to get the Dying GASP Config.

#### Parameters

<i>OUT</i>	<p>pDestSMSNum[OUT]</p> <ul style="list-style-type: none"> <li>SMS Destination Number as string of 8 bit ASCII Characters Max 20 chars.</li> <li>Optional parameter.</li> </ul>
<i>OUT</i>	<p>pDestSMSContent[OUT]</p> <ul style="list-style-type: none"> <li>SMS Content as a string of 8 bit ASCII text characters Max 160 chars.</li> <li>Optional parameter.</li> </ul>

### 8.662.2 Field Documentation

8.662.2.1 **BYTE\*** setDyingGaspCfg::pDestSMSContent

8.662.2.2 **BYTE\*** setDyingGaspCfg::pDestSMSNum

## 8.663 SetIMSSMSConfigReq Struct Reference

#### Data Fields

- BYTE \*** pSMSFormat
- BYTE \*** pSMSOverIPNwInd
- BYTE \*** pPhoneCtxtURLen
- BYTE \*** pPhoneCtxtURI

### 8.663.1 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>

## 8.663.2 Field Documentation

8.663.2.1 **BYTE\*** SetIMSSMSConfigReq::pPhoneCtxtURI

8.663.2.2 **BYTE\*** SetIMSSMSConfigReq::pPhoneCtxtURLen

8.663.2.3 **BYTE\*** SetIMSSMSConfigReq::pSMSFormat

8.663.2.4 **BYTE\*** SetIMSSMSConfigReq::pSMSOverIPNwInd

## 8.664 SetIMSSMSConfigResp Struct Reference

### Data Fields

- **BYTE \*** [pSettingResp](#)

### 8.664.1 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>
---------------------	---

## 8.664.2 Field Documentation

8.664.2.1 **BYTE\*** SetIMSSMSConfigResp::pSettingResp

## 8.665 SetIMSUserConfigReq Struct Reference

### Data Fields

- **BYTE \*** [pIMSDomainLen](#)
- **BYTE \*** [pIMSDomain](#)

### 8.665.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig request parameters.

#### Parameters

<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> </ul>

## 8.665.2 Field Documentation

8.665.2.1 **BYTE\*** SetIMSUserConfigReq::pIMSDomain

8.665.2.2 **BYTE\*** SetIMSUserConfigReq::pIMSDomainLen

## 8.666 SetIMSUserConfigResp Struct Reference

### Data Fields

- **BYTE \*** pSettingResp

### 8.666.1 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>
---------------------	---

### 8.666.2 Field Documentation

8.666.2.1 **BYTE\*** SetIMSUserConfigResp::pSettingResp

## 8.667 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

### 8.667.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig request parameters.

#### 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 - 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>

## 8.667.2 Field Documentation

8.667.2.1 **BYTE\*** SetIMSVolIPConfigReq::pAmrMode

8.667.2.2 **BYTE\*** SetIMSVolIPConfigReq::pAmrOctetAligned

8.667.2.3 **BYTE\*** SetIMSVolIPConfigReq::pAmrWbEnable

8.667.2.4 **WORD\*** SetIMSVolIPConfigReq::pAmrWBMode

8.667.2.5 **BYTE\*** SetIMSVolIPConfigReq::pAmrWBOctetAligned

8.667.2.6 **WORD\*** SetIMSVolIPConfigReq::pMinSessionExpiryTimer

8.667.2.7 **WORD\*** SetIMSVolIPConfigReq::pRingBackTimer

8.667.2.8 **WORD\*** SetIMSVolIPConfigReq::pRingingTimer

8.667.2.9 **WORD\*** SetIMSVolIPConfigReq::pRTPRTCPInactTimer

8.667.2.10 **BYTE\*** SetIMSVolIPConfigReq::pScrAmrEnable

8.667.2.11 **BYTE\*** SetIMSVolIPConfigReq::pScrAmrWbEnable

8.667.2.12 **WORD\*** SetIMSVolIPConfigReq::pSessionExpiryTimer

## 8.668 SetIMSVolIPConfigResp Struct Reference

### Data Fields

- BYTE \*** [pSettingResp](#)

### 8.668.1 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>
---------------------	---

### 8.668.2 Field Documentation

8.668.2.1 **BYTE\*** SetIMSVoIPConfigResp::pSettingResp

## 8.669 SetM2MAudioAVCFGReq Struct Reference

#### Data Fields

- [BYTE Profile](#)
- [BYTE Device](#)
- [BYTE PIFACEId](#)
- [PCMparams \\* pPCMPParams](#)

### 8.669.1 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>

### 8.669.2 Field Documentation

8.669.2.1 **BYTE** SetM2MAudioAVCFGReq::Device

8.669.2.2 **BYTE** SetM2MAudioAVCFGReq::PIFACEId

8.669.2.3 PCMparams\* SetM2MAudioAVCFGReq::pPCMParams

8.669.2.4 BYTE SetM2MAudioAVCFGReq::Profile

## 8.670 SetM2MAudioLPBKReq Struct Reference

### Data Fields

- [BYTE Enable](#)

### 8.670.1 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>
---------------	--

### 8.670.2 Field Documentation

8.670.2.1 BYTE SetM2MAudioLPBKReq::Enable

## 8.671 SetM2MAudioProfileReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE \\* pEarMute](#)
- [BYTE \\* pMicMute](#)
- [BYTE \\* pGenerator](#)
- [BYTE \\* pVolume](#)
- [BYTE \\* pCwtMute](#)

### 8.671.1 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>

## 8.671.2 Field Documentation

8.671.2.1 **BYTE\*** SetM2MAudioProfileReq::pCwtMute

8.671.2.2 **BYTE\*** SetM2MAudioProfileReq::pEarMute

8.671.2.3 **BYTE\*** SetM2MAudioProfileReq::pGenerator

8.671.2.4 **BYTE\*** SetM2MAudioProfileReq::pMicMute

8.671.2.5 **BYTE** SetM2MAudioProfileReq::Profile

8.671.2.6 **BYTE\*** SetM2MAudioProfileReq::pVolume

## 8.672 SetM2MAudioVolumeReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE Generator](#)
- [BYTE Level](#)

### 8.672.1 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>

## 8.672.2 Field Documentation

8.672.2.1 BYTE SetM2MAudioVolumeReq::Generator

8.672.2.2 BYTE SetM2MAudioVolumeReq::Level

8.672.2.3 BYTE SetM2MAudioVolumeReq::Profile

## 8.673 SetM2MAVMuteReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [BYTE EarMute](#)
- [BYTE MicMute](#)
- [BYTE \\* pCwtMute](#)

### 8.673.1 Detailed Description

This structure contains the SLQSSetM2MAVMute 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>

## 8.673.2 Field Documentation

8.673.2.1 **BYTE** SetM2MAVMuteReq::EarMute

8.673.2.2 **BYTE** SetM2MAVMuteReq::MicMute

8.673.2.3 **BYTE\*** SetM2MAVMuteReq::pCwtMute

8.673.2.4 **BYTE** SetM2MAVMuteReq::Profile

## 8.674 SetM2MSpkrGainReq Struct Reference

### Data Fields

- [BYTE Profile](#)
- [WORD Value](#)

### 8.674.1 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>

### 8.674.2 Field Documentation

8.674.2.1 **BYTE** SetM2MSpkrGainReq::Profile

8.674.2.2 **WORD** SetM2MSpkrGainReq::Value

## 8.675 setPINProtection Struct Reference

### Data Fields

- [BYTE pinID](#)
- [BYTE pinOperation](#)
- [BYTE pinLength](#)
- [BYTE pinValue](#) [255]

### 8.675.1 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>

## 8.675.2 Field Documentation

8.675.2.1 BYTE setPINProtection::pinID

8.675.2.2 BYTE setPINProtection::pinLength

8.675.2.3 BYTE setPINProtection::pinOperation

8.675.2.4 BYTE setPINProtection::pinValue[255]

## 8.676 SetRegMgrConfigReq Struct Reference

## Data Fields

- WORD \* pPriCSCFPort
- BYTE \* pCSCFPortNameLen
- BYTE \* pCSCFPortName
- BYTE \* pIMSTestMode

## 8.676.1 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>pCSCFPortNameLen</i>	<ul style="list-style-type: none"> <li>Length of the CSCF Port 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>

## 8.676.2 Field Documentation

8.676.2.1 **BYTE\*** SetRegMgrConfigReq::pCSCFPortName

8.676.2.2 **BYTE\*** SetRegMgrConfigReq::pCSCFPortNameLen

8.676.2.3 **BYTE\*** SetRegMgrConfigReq::pIMSTestMode

8.676.2.4 **WORD\*** SetRegMgrConfigReq::pPriCSCFPort

## 8.677 SetRegMgrConfigResp Struct Reference

### Data Fields

- **BYTE \*** [pSettingResp](#)

### 8.677.1 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>
---------------------	---

## 8.677.2 Field Documentation

8.677.2.1 **BYTE\*** SetRegMgrConfigResp::pSettingResp

## 8.678 setSignalStrengthInfo Struct Reference

### Data Fields

- **CDMARSSIThresh \*** [pCDMARSSIThresh](#)
- **WORD \*** [pCDMARSSIDelta](#)
- **CDMAECIOThresh \*** [pCDMAECIOThresh](#)
- **WORD \*** [pCDMAECIODelta](#)
- **HDRRSSIThresh \*** [pHDRRSSIThresh](#)
- **WORD \*** [pHDRRSSIDelta](#)

- HDRECIOTresh \* pHRECIOTresh
- WORD \* pHRECIODelta
- HDRSINRThreshold \* pHDRSINRThresh
- WORD \* pHDRSINRDelta
- HDRIOTresh \* pHRIOTresh
- WORD \* pHRIODelta
- GSMRSSITresh \* pGSMRSSITresh
- WORD \* pGSMRSSIDelta
- WCDMARSSITresh \* pWCDMARSSITresh
- WORD \* pWCDMARSSIDelta
- WCDMAECIOTresh \* pWCDMAECIOTresh
- WORD \* pWCDMAECIODelta
- LTERSSITresh \* pLTERSSITresh
- WORD \* pLTERSSIDelta
- LTESNRThreshold \* pLTESNRThresh
- WORD \* pLTESNRDelta
- LTERSRQThresh \* pLTERSRQThresh
- WORD \* pLTERSRQDelta
- LTERSRPThresh \* pLTERSRPThresh
- WORD \* pLTERSRPDelta
- LTESigRptConfig \* pLTESigRptConfig
- TDSCDMARSCPTresh \* pTDSCDMARSCPTresh
- WORD \* pTDSCDMARSCPDelta
- TDSCDMARSSITresh \* pTDSCDMARSSITresh
- ULONG \* pTDSCDMARSSIDelta
- TDSCDMAECIOTresh \* pTDSCDMAECIOTresh
- ULONG \* pTDSCDMAECIODelta
- TDSCDMASINRThresh \* pTDSCDMASINRThresh
- ULONG \* pTDSCDMASINRDelta

### 8.678.1 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="#">CDMARSSITresh</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="#">CDMAECIOTresh</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>pHRRSSI- Thresh</i>	<ul style="list-style-type: none"> <li>HDR RSSI Threshold List</li> <li>See <a href="#">HRRSSIThresh</a> for more details</li> </ul>
<i>pHRRSSIDelta</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>pHRECIO- Thresh</i>	<ul style="list-style-type: none"> <li>HDR ECIO Threshold List</li> <li>See <a href="#">HRECIOThresh</a> for more details</li> </ul>
<i>pHRECIODelta</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>pHRSINR- Thresh</i>	<ul style="list-style-type: none"> <li>HDR SINR Threshold List</li> <li>See <a href="#">HRSINRThreshold</a> for more details</li> </ul>
<i>pHRSINRDelta</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>pLTERSNR-Thresh</i>	<ul style="list-style-type: none"> <li>• LTE SNR Threshold List</li> <li>• See <a href="#">LTERSNRThreshold</a> for more details</li> </ul>
<i>pLTERSNRDelta</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>pLTERSRQ-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-OTresh</i>	<ul style="list-style-type: none"> <li>• TDSCDMA ECIO Threshold List</li> <li>• See <a href="#">TDSCDMAECIOTresh</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>

## 8.678.2 Field Documentation

8.678.2.1 **WORD\*** `setSignalStrengthInfo::pCDMAECIODelta`

8.678.2.2 **CDMAECIOTresh\*** `setSignalStrengthInfo::pCDMAECIOTresh`

8.678.2.3 **WORD\*** `setSignalStrengthInfo::pCDMARSSIDelta`

8.678.2.4 **CDMARSSITresh\*** `setSignalStrengthInfo::pCDMARSSITresh`

8.678.2.5 **WORD\*** `setSignalStrengthInfo::pGSMRSSIDelta`

8.678.2.6 **GSMRSSITresh\*** `setSignalStrengthInfo::pGSMRSSITresh`

8.678.2.7 **WORD\*** `setSignalStrengthInfo::pHDRECIODelta`

8.678.2.8 **HDRECIOTresh\*** `setSignalStrengthInfo::pHDRECIOTresh`

8.678.2.9 **WORD\*** `setSignalStrengthInfo::pHDRIODelta`

8.678.2.10 **HDRIOTresh\*** `setSignalStrengthInfo::pHDRIOTresh`

8.678.2.11 **WORD\*** `setSignalStrengthInfo::pHRRSSIDelta`

8.678.2.12 **HDRRSSITresh\*** `setSignalStrengthInfo::pHRRSSITresh`

8.678.2.13 **WORD\*** `setSignalStrengthInfo::pHRSINRDelta`

8.678.2.14 **HDRSINRThreshold\*** `setSignalStrengthInfo::pHRSINRThresh`

8.678.2.15 **WORD\*** `setSignalStrengthInfo::pLTERSRPDelta`

8.678.2.16 **LTERSRPThresh\*** `setSignalStrengthInfo::pLTERSRPThresh`

8.678.2.17 **WORD\*** `setSignalStrengthInfo::pLTERSQRQDelta`

- 8.678.2.18 **LTERSQRThresh**\* setSignalStrengthInfo::pLTERSQRThresh
- 8.678.2.19 **WORD**\* setSignalStrengthInfo::pLTERSSIDelta
- 8.678.2.20 **LTERSSIThresh**\* setSignalStrengthInfo::pLTERSSIThresh
- 8.678.2.21 **LTESigRptConfig**\* setSignalStrengthInfo::pLTESigRptConfig
- 8.678.2.22 **WORD**\* setSignalStrengthInfo::pLTESNRDelta
- 8.678.2.23 **LTESNRThreshold**\* setSignalStrengthInfo::pLTESNRThresh
- 8.678.2.24 **ULONG**\* setSignalStrengthInfo::pTDSCDMAECIODelta
- 8.678.2.25 **TDSCDMAECIOThresh**\* setSignalStrengthInfo::pTDSCDMAECIOThresh
- 8.678.2.26 **WORD**\* setSignalStrengthInfo::pTDSCDMARSCPDelta
- 8.678.2.27 **TDSCDMARSCPThresh**\* setSignalStrengthInfo::pTDSCDMARSCPThresh
- 8.678.2.28 **ULONG**\* setSignalStrengthInfo::pTDSCDMARSSIDelta
- 8.678.2.29 **TDSCDMARSSIThresh**\* setSignalStrengthInfo::pTDSCDMARSSIThresh
- 8.678.2.30 **ULONG**\* setSignalStrengthInfo::pTDSCDMASINRDelta
- 8.678.2.31 **TDSCDMASINRThresh**\* setSignalStrengthInfo::pTDSCDMASINRThresh
- 8.678.2.32 **WORD**\* setSignalStrengthInfo::pWCDMAECIODelta
- 8.678.2.33 **WCDMAECIOThresh**\* setSignalStrengthInfo::pWCDMAECIOThresh
- 8.678.2.34 **WORD**\* setSignalStrengthInfo::pWCDMARSSIDelta
- 8.678.2.35 **WCDMARSSIThresh**\* setSignalStrengthInfo::pWCDMARSSIThresh

## 8.679 SetSIPConfigReq Struct Reference

### Data Fields

- **WORD** \* pSIPLocalPort
- **ULONG** \* pTimerSIPReg
- **ULONG** \* pSubscribeTimer
- **ULONG** \* pTimerT1
- **ULONG** \* pTimerT2
- **ULONG** \* pTimerTf
- **BYTE** \* pSigCompEnabled

### 8.679.1 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>

## 8.679.2 Field Documentation

8.679.2.1 **BYTE\*** SetSIPConfigReq::pSigCompEnabled8.679.2.2 **WORD\*** SetSIPConfigReq::pSIPLocalPort8.679.2.3 **ULONG\*** SetSIPConfigReq::pSubscribeTimer8.679.2.4 **ULONG\*** SetSIPConfigReq::pTimerSIPReg8.679.2.5 **ULONG\*** SetSIPConfigReq::pTimerT18.679.2.6 **ULONG\*** SetSIPConfigReq::pTimerT28.679.2.7 **ULONG\*** SetSIPConfigReq::pTimerTf

## 8.680 SetSIPConfigResp Struct Reference

## Data Fields

- BYTE \*** [pSettingResp](#)

## 8.680.1 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>
---------------------	---

## 8.680.2 Field Documentation

8.680.2.1 BYTE\* SetSIPConfigResp::pSettingResp

## 8.681 sGetDeviceSeriesResult Struct Reference

## Data Fields

- enum [eGobiDeviceSeries](#) [eDevice](#)
- ULONG [uResult](#)

## 8.681.1 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

## 8.681.2 Field Documentation

8.681.2.1 enum [eGobiDeviceSeries](#) [sGetDeviceSeriesResult::eDevice](#)8.681.2.2 ULONG [sGetDeviceSeriesResult::uResult](#)

## 8.682 sidNid Struct Reference

## Data Fields

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

## 8.682.1 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>

## 8.682.2 Field Documentation

8.682.2.1 WORD sidNid::nid

8.682.2.2 WORD sidNid::sid

## 8.683 sigInfo Struct Reference

### Data Fields

- [RSSIThresh](#) \* [pRSSIThresh](#)
- [ECIOThresh](#) \* [pECIOThresh](#)
- [HDRSINRThresh](#) \* [pHDRSINRThresh](#)
- [LTESNRThresh](#) \* [pLTESNRThresh](#)
- [IOTresh](#) \* [pIOTresh](#)
- [RSRQThresh](#) \* [pRSRQThresh](#)
- [RSRPThresh](#) \* [pRSRPThresh](#)
- [LTESigRptCfg](#) \* [pLTESigRptCfg](#)
- [TDSCDMASINRCONFTresh](#) \* [pTDSCDMASINRCONFTresh](#)

### 8.683.1 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>pECIOThresh</i>	<ul style="list-style-type: none"> <li>• ECIO Threshold List</li> <li>• See <a href="#">ECIOThresh</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>pIOTresh</i>	<ul style="list-style-type: none"> <li>• IO Threshold List</li> <li>• See <a href="#">IOTresh</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>
<i>pTDSCDMASINRCONFTresh</i>	<ul style="list-style-type: none"> <li>• TD-SCDMA SINR Threshold List</li> <li>• See <a href="#">TDSCDMASINRCONFTresh</a> for more details</li> </ul>

## 8.683.2 Field Documentation

8.683.2.1 **ECIOThresh\*** sigInfo::pECIOThresh

8.683.2.2 **HDRSINRThresh\*** sigInfo::pHDRSINRThresh

8.683.2.3 **IOThresh\*** sigInfo::pIOThresh

8.683.2.4 **LTESigRptCfg\*** sigInfo::pLTESigRptCfg

8.683.2.5 **LTESNRThresh\*** sigInfo::pLTESNRThresh

8.683.2.6 **RSRPThresh\*** sigInfo::pRSRPThresh

8.683.2.7 **RSRQThresh\*** sigInfo::pRSRQThresh

8.683.2.8 **RSSIThresh\*** sigInfo::pRSSIThresh

8.683.2.9 **TDSCDMASINRCONFTresh\*** sigInfo::pTDSCDMASINRCONFTresh

## 8.684 signalInfo Struct Reference

### Data Fields

- [BYTE](#) signalType
- [BYTE](#) alertPitch
- [BYTE](#) signal

### 8.684.1 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>

## 8.684.2 Field Documentation

8.684.2.1 BYTE signalInfo::alertPitch

8.684.2.2 BYTE signalInfo::signal

8.684.2.3 BYTE signalInfo::signalType

## 8.685 SignalStrengthDataType Struct Reference

### Data Fields

- [BYTE thresholdsSize](#)
- [INT8 thresholds](#) [5]

## 8.685.1 Field Documentation

8.685.1.1 INT8 SignalStrengthDataType::thresholds[5]

8.685.1.2 BYTE SignalStrengthDataType::thresholdsSize

## 8.686 slot\_t Struct Reference

### Data Fields

- uint32\_t [uPhyCardStatus](#)
- uint32\_t [uPhySlotStatus](#)
- uint8\_t [bLogicalSlot](#)
- uint8\_t [bCCIDLength](#)
- uint8\_t [bCCID](#) [255]

## 8.686.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

### Parameters

<i>uPhyCardStatus</i>	<ul style="list-style-type: none"> <li>• State of the card in the Pyhsical Slot Status. <ul style="list-style-type: none"> <li>– 0x00 - Unknown.</li> <li>– 0x01 - Absent.</li> <li>– 0x02 - Present.</li> </ul> </li> </ul>
<i>uPhySlotStatus</i>	<ul style="list-style-type: none"> <li>• State of the Physical Slot status. <ul style="list-style-type: none"> <li>– 0x00 Inactive.</li> <li>– 0x01 Activate.</li> </ul> </li> </ul>

<i>bLogicalSlot</i>	<ul style="list-style-type: none"> <li>• Logical Slot associated with this physical slot. This is valid if the physical slot is active. <ul style="list-style-type: none"> <li>– 1 - Slot 1.</li> <li>– 2 - Slot 2.</li> <li>– 3 - Slot 3.</li> <li>– 4 - Slot 4.</li> <li>– 5 - Slot 5.</li> </ul> </li> </ul>
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> <li>• Number of sets the sets of ICCID</li> </ul>
<i>bICCID[MAX_ICCID_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Contains the ICCID of the card in the physical slot.</li> </ul>

## 8.686.2 Field Documentation

8.686.2.1 `uint8_t slot_t::bICCID[255]`

8.686.2.2 `uint8_t slot_t::bICCIDLength`

8.686.2.3 `uint8_t slot_t::bLogicalSlot`

8.686.2.4 `uint32_t slot_t::uPhyCardStatus`

8.686.2.5 `uint32_t slot_t::uPhySlotStatus`

## 8.687 slotInf Struct Reference

### Data Fields

- `uint8_t cardState`
- `uint8_t upinState`
- `uint8_t upinRetries`
- `uint8_t upukRetries`
- `uint8_t errorState`
- `uint8_t numApp`
- `appStats AppStatus [10]`

### 8.687.1 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_APPLICATIONS]</i>	<ul style="list-style-type: none"> <li>See <a href="#">appStats</a> for more information.</li> </ul>

## 8.687.2 Field Documentation

### 8.687.2.1 appStats slotInf::AppStatus[10]

### 8.687.2.2 uint8\_t slotInf::cardState

### 8.687.2.3 uint8\_t slotInf::errorState

### 8.687.2.4 uint8\_t slotInf::numApp

8.687.2.5 `uint8_t slotInf::upinRetries`

8.687.2.6 `uint8_t slotInf::upinState`

8.687.2.7 `uint8_t slotInf::upukRetries`

## 8.688 slotInfo Struct Reference

### Data Fields

- [BYTE cardState](#)
- [BYTE upinState](#)
- [BYTE upinRetries](#)
- [BYTE upukRetries](#)
- [BYTE errorState](#)
- [BYTE numApp](#)
- [appStatus AppStatus](#) [10]

### 8.688.1 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_APPLICATIONS]</i>	<ul style="list-style-type: none"> <li>See <a href="#">appStatus</a> for more information.</li> </ul>

## 8.688.2 Field Documentation

### 8.688.2.1 appStatus slotInfo::AppStatus[10]

### 8.688.2.2 BYTE slotInfo::cardState

### 8.688.2.3 BYTE slotInfo::errorState

### 8.688.2.4 BYTE slotInfo::numApp

### 8.688.2.5 BYTE slotInfo::upinRetries

### 8.688.2.6 BYTE slotInfo::upinState

### 8.688.2.7 BYTE slotInfo::upukRetries

## 8.689 slots\_t Struct Reference

### Data Fields

- [slot\\_t uimSlotStatus](#) [255]

### 8.689.1 Field Documentation

#### 8.689.1.1 slot\_t slots\_t::uimSlotStatus[255]

## 8.690 slqsautoconnect Struct Reference

## Data Fields

- [BOOL action](#)
- [BYTE acsetting](#)
- [BYTE acroamsetting](#)

### 8.690.1 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>

### 8.690.2 Field Documentation

8.690.2.1 **BYTE** slqsautoconnect::acroamsetting

8.690.2.2 **BYTE** slqsautoconnect::acsetting

8.690.2.3 **BOOL** slqsautoconnect::action

## 8.691 SLQSDeleteProfileParams Struct Reference

## Data Fields

- [BYTE profileType](#)
- [BYTE profileIndex](#)

### 8.691.1 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 between 1 - 16 (EM/MC73xx or earlier)</li> <li>– Value between 1 - 24 (EM/MC74xx onwards)</li> </ul> </li> </ul>
---------------------	--

## 8.691.2 Field Documentation

### 8.691.2.1 BYTE SLQSDDeleteProfileParams::profileIndex

### 8.691.2.2 BYTE SLQSDDeleteProfileParams::profileType

## 8.692 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]

### 8.692.1 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> <li>• deprecated on EM/MC74xx(9x30) devices</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>

## 8.692.2 Field Documentation

8.692.2.1 CHAR slqsfwinfo\_s::appversion\_str[85]

8.692.2.2 CHAR slqsfwinfo\_s::bootversion\_str[85]

8.692.2.3 CHAR slqsfwinfo\_s::carrier\_str[20]

8.692.2.4 CHAR slqsfwinfo\_s::cur\_carr\_name[17]

8.692.2.5 CHAR slqsfwinfo\_s::cur\_carr\_rev[13]

8.692.2.6 CHAR slqsfwinfo\_s::modelid\_str[20]

8.692.2.7 CHAR slqsfwinfo\_s::packageid\_str[85]

8.692.2.8 CHAR slqsfwinfo\_s::priversion\_str[16]

8.692.2.9 CHAR slqsfwinfo\_s::sku\_str[15]

## 8.693 SlqsNas3GppNetworkInfo Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [ULONG InUse](#)
- [ULONG Roaming](#)
- [ULONG Forbidden](#)
- [ULONG Preferred](#)
- [CHAR Description](#) [255]

### 8.693.1 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>

## 8.693.2 Field Documentation

8.693.2.1 **CHAR** SIqsNas3GppNetworkInfo::Description[255]

8.693.2.2 **ULONG** SIqsNas3GppNetworkInfo::Forbidden

8.693.2.3 **ULONG** SIqsNas3GppNetworkInfo::InUse

8.693.2.4 **WORD** SIqsNas3GppNetworkInfo::MCC

8.693.2.5 **WORD** SIqsNas3GppNetworkInfo::MNC

8.693.2.6 **ULONG** SIqsNas3GppNetworkInfo::Preferred

8.693.2.7 **ULONG** SIqsNas3GppNetworkInfo::Roaming

## 8.694 SIqsNasPcsDigit Struct Reference

### Data Fields

- [WORD MCC](#)
- [WORD MNC](#)
- [BYTE includes\\_pcs\\_digit](#)

### 8.694.1 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>

### 8.694.2 Field Documentation

8.694.2.1 **BYTE** SlqsNasPcsDigit::includes\_pcs\_digit

8.694.2.2 **WORD** SlqsNasPcsDigit::MCC

8.694.2.3 **WORD** SlqsNasPcsDigit::MNC

## 8.695 slqssendasyncsmsparams\_s Struct Reference

### Data Fields

- [ULONG messageFormat](#)
- [ULONG messageSize](#)
- [BYTE \\* pMessage](#)
- [BYTE \\* pForceOnDC](#)
- [BYTE \\* pServiceOption](#)
- [BYTE \\* pFollowOnDC](#)
- [BYTE \\* pLinktimer](#)
- [BYTE \\* pSmsOnIms](#)
- [BYTE \\* pRetryMessage](#)
- [ULONG \\* pRetryMessageld](#)
- [ULONG \\* pUserData](#)

### 8.695.1 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>
------------------	---

## 8.695.2 Field Documentation

- 8.695.2.1 **ULONG** slqssendasynsmsparams\_s::messageFormat
- 8.695.2.2 **ULONG** slqssendasynsmsparams\_s::messageSize
- 8.695.2.3 **BYTE\*** slqssendasynsmsparams\_s::pFollowOnDC
- 8.695.2.4 **BYTE\*** slqssendasynsmsparams\_s::pForceOnDC
- 8.695.2.5 **BYTE\*** slqssendasynsmsparams\_s::pLinktimer
- 8.695.2.6 **BYTE\*** slqssendasynsmsparams\_s::pMessage
- 8.695.2.7 **BYTE\*** slqssendasynsmsparams\_s::pRetryMessage
- 8.695.2.8 **ULONG\*** slqssendasynsmsparams\_s::pRetryMessageId
- 8.695.2.9 **BYTE\*** slqssendasynsmsparams\_s::pServiceOption
- 8.695.2.10 **BYTE\*** slqssendasynsmsparams\_s::pSmsOnlms
- 8.695.2.11 **ULONG\*** slqssendasynsmsparams\_s::pUserData

## 8.696 slqssendsmsparams\_s Struct Reference

### Data Fields

- [ULONG messageFormat](#)
- [ULONG messageSize](#)
- [BYTE \\* pMessage](#)
- [USHORT messageId](#)
- [ULONG messageFailureCode](#)
- [BYTE \\* pLinktimer](#)
- [BYTE \\* pSmsOnlms](#)

### 8.696.1 Detailed Description

This structure contains SMS parameters

## Parameters

<i>message-Format</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>• The length of the message contents in bytes</li> </ul>
<i>pMessage</i> [IN]	<ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>
<i>pMessageID</i> [OUT]	<ul style="list-style-type: none"> <li>• message reference ID</li> </ul>
<i>pMessage-FailureCode</i> [OUT]	<ul style="list-style-type: none"> <li>• (Optional) Message Failure Code</li> <li>• If cause code is not provided, then value will be 0xFFFFFFFF</li> </ul>
<i>pSmsOnIms</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) SMS on IMS</li> <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>pLinktimer</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) Link Timer</li> <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>

## 8.696.2 Field Documentation

8.696.2.1 ULONG slqssendsmsparams\_s::messageFailureCode

8.696.2.2 ULONG slqssendsmsparams\_s::messageFormat

8.696.2.3 USHORT slqssendsmsparams\_s::messageID

8.696.2.4 ULONG slqssendsmsparams\_s::messageSize

8.696.2.5 BYTE\* slqssendsmsparams\_s::pLinktimer

8.696.2.6 BYTE\* slqssendsmsparams\_s::pMessage

8.696.2.7 BYTE\* slqssendsmsparams\_s::pSmsOnIms

## 8.697 slqsSessionStateInfo Struct Reference

## Data Fields

- [qaQmiInterfaceInfo](#) \* [pQmiInterfaceInfo](#)
- [ULONG](#) [reconfiguration\\_required](#)
- [ULONG](#) [state](#)
- [ULONG](#) [sessionEndReason](#)

## 8.697.1 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>

## 8.697.2 Field Documentation

8.697.2.1 [qaQmiInterfaceInfo](#)\* [slqsSessionStateInfo::pQmiInterfaceInfo](#)

8.697.2.2 [ULONG](#) [slqsSessionStateInfo::reconfiguration\\_required](#)

8.697.2.3 [ULONG](#) [slqsSessionStateInfo::sessionEndReason](#)

8.697.2.4 [ULONG](#) [slqsSessionStateInfo::state](#)

## 8.698 slqsSignalStrengthInfo Struct Reference

## Data Fields

- [USHORT](#) [signalStrengthReqMask](#)
- [USHORT](#) [rxSignalStrengthListLen](#)
- struct [rxSignalStrengthListElement](#) [rxSignalStrengthList](#) [18]
- [USHORT](#) [ecioListLen](#)
- struct [ecioListElement](#) [ecioList](#) [18]
- [INT32](#) [Io](#)
- [BYTE](#) [sinr](#)

- [USHORT errorRateListLen](#)
- struct [errorRateListElement errorRateList](#) [18]
- struct [rsrqInformation rsrqInfo](#)
- [SHORT ltesnr](#)
- [SHORT ltersrp](#)

### 8.698.1 Detailed Description

This structure contains the Signal Strength Information

#### Parameters

<i>signalStrength-ReqMask</i> [IN]	<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>rxSignalStrengthListLen</i> [OUT]	<ul style="list-style-type: none"> <li>• Number of elements in Receive Signal Strength List</li> </ul>
<i>rxSignalStrengthList</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">rxSignalStrengthListElement</a> for more information</li> </ul>
<i>ecioListLen</i> [OUT]	<ul style="list-style-type: none"> <li>• Number of elements in ECIO List</li> </ul>

<i>ecioList</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">ecioListElement</a> for more information</li> </ul>
<i>lo</i> [OUT]	<ul style="list-style-type: none"> <li>Received lo in dBm; IO is only applicable for 1xEV-DO</li> </ul>
<i>sinr</i> [OUT]	<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               <ul style="list-style-type: none"> <li>SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3</li> <li>SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 -</li> <li>SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB</li> </ul> </li> </ul> </li> </ul>
<i>errorRateListLen</i> [OUT]	<ul style="list-style-type: none"> <li>Number of elements in Error Rate List</li> </ul>
<i>errorRateList</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">errorRateListElement</a> for more information</li> </ul>
<i>rsrqInfo</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">rsrqInformation</a> for more information</li> </ul>
<i>ltesnr</i> [OUT]	<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> [OUT]	<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>

## 8.698.2 Field Documentation

8.698.2.1 struct `ecioListElement` `slqsSignalStrengthInfo::ecioList`[18]

8.698.2.2 USHORT `slqsSignalStrengthInfo::ecioListLen`

8.698.2.3 struct `errorRateListElement` `slqsSignalStrengthInfo::errorRateList`[18]

8.698.2.4 USHORT `slqsSignalStrengthInfo::errorRateListLen`

8.698.2.5 INT32 `slqsSignalStrengthInfo::lo`

8.698.2.6 SHORT `slqsSignalStrengthInfo::ltersrp`

8.698.2.7 SHORT `slqsSignalStrengthInfo::ltesnr`

8.698.2.8 struct `rsrqInformation` `slqsSignalStrengthInfo::rsrqInfo`

8.698.2.9 struct `rxSignalStrengthListElement` `slqsSignalStrengthInfo::rxSignalStrengthList`[18]

8.698.2.10 USHORT `slqsSignalStrengthInfo::rxSignalStrengthListLen`

8.698.2.11 USHORT slqsSignalStrengthInfo::signalStrengthReqMask

8.698.2.12 BYTE slqsSignalStrengthInfo::sinr

## 8.699 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](#)

### 8.699.1 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>ecioThresholdListLen</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 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>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>lteresrpdelta</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

## 8.699.2 Field Documentation

- 8.699.2.1 BYTE SLQSSignalStrengthsIndReq::ecioDelta
- 8.699.2.2 SHORT SLQSSignalStrengthsIndReq::ecioThresholdList[10]
- 8.699.2.3 BYTE SLQSSignalStrengthsIndReq::ecioThresholdListLen
- 8.699.2.4 BYTE SLQSSignalStrengthsIndReq::ioDelta
- 8.699.2.5 BYTE SLQSSignalStrengthsIndReq::lteRsrpDelta
- 8.699.2.6 WORD SLQSSignalStrengthsIndReq::lteSnrDelta
- 8.699.2.7 BYTE SLQSSignalStrengthsIndReq::rsrqDelta
- 8.699.2.8 BYTE SLQSSignalStrengthsIndReq::rxSignalStrengthDelta
- 8.699.2.9 BYTE SLQSSignalStrengthsIndReq::sinrDelta
- 8.699.2.10 BYTE SLQSSignalStrengthsIndReq::sinrThresholdList[5]
- 8.699.2.11 BYTE SLQSSignalStrengthsIndReq::sinrThresholdListLen

## 8.700 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](#)

### 8.700.1 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               <ul style="list-style-type: none"> <li>- SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3</li> <li>- SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 -</li> <li>SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB</li> </ul> </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

### 8.700.2 Field Documentation

- 8.700.2.1 struct ecioListElement SLQSSignalStrengthsInformation::ecioInfo
- 8.700.2.2 struct errorRateListElement SLQSSignalStrengthsInformation::errorRateInfo
- 8.700.2.3 ULONG SLQSSignalStrengthsInformation::io
- 8.700.2.4 struct lteRsrpInformation SLQSSignalStrengthsInformation::lteRsrpinfo
- 8.700.2.5 struct lteSnrinformation SLQSSignalStrengthsInformation::lteSnrininfo
- 8.700.2.6 struct rsrqInformation SLQSSignalStrengthsInformation::rsrqInfo
- 8.700.2.7 struct rxSignalStrengthListElement SLQSSignalStrengthsInformation::rxSignalStrengthInfo
- 8.700.2.8 BYTE SLQSSignalStrengthsInformation::sinr

## 8.701 slqsWdsEventInfo Struct Reference

### Data Fields

- [qaQmiInterfaceInfo](#) \* [pQmiInterfaceInfo](#)
- [ULONG](#) \* [pDormancyStatus](#)
- [ULONG](#) \* [pDataBearer](#)
- [ULONG](#) \* [pPacketsCountTX](#)
- [ULONG](#) \* [pPacketsCountRX](#)
- [ULONGLONG](#) \* [pTotalBytesTX](#)
- [ULONGLONG](#) \* [pTotalBytesRX](#)

### 8.701.1 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>

## 8.701.2 Field Documentation

8.701.2.1 **ULONG\*** `slqsWdsEventInfo::pDataBearer`8.701.2.2 **ULONG\*** `slqsWdsEventInfo::pDormancyStatus`8.701.2.3 **ULONG\*** `slqsWdsEventInfo::pPacketsCountRX`8.701.2.4 **ULONG\*** `slqsWdsEventInfo::pPacketsCountTX`

8.701.2.5 `qaQmiInterfaceInfo*` `slqsWdsEventInfo::pQmiInterfaceInfo`8.701.2.6 `ULONGLONG*` `slqsWdsEventInfo::pTotalBytesRX`8.701.2.7 `ULONGLONG*` `slqsWdsEventInfo::pTotalBytesTX`

## 8.702 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`

### 8.702.1 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>

## 8.702.2 Field Documentation

8.702.2.1 BYTE SMSAsyncRawSend\_s::alphaIDLen

8.702.2.2 WORD SMSAsyncRawSend\_s::causeCode

8.702.2.3 BYTE SMSAsyncRawSend\_s::errorClass

8.702.2.4 WORD SMSAsyncRawSend\_s::messageID

8.702.2.5 BYTE SMSAsyncRawSend\_s::msgDelFailureCause

8.702.2.6 BYTE SMSAsyncRawSend\_s::msgDelFailureType

8.702.2.7 BYTE\* SMSAsyncRawSend\_s::pAlphaID

8.702.2.8 WORD SMSAsyncRawSend\_s::RPCause

8.702.2.9 WORD SMSAsyncRawSend\_s::sendStatus

8.702.2.10 BYTE SMSAsyncRawSend\_s::TPCause

8.702.2.11 ULONG SMSAsyncRawSend\_s::userData

## 8.703 sMSCAddress Struct Reference

### Data Fields

- uint8\_t [length](#)
- uint8\_t [data](#) [256]

### 8.703.1 Detailed Description

#### 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>

### 8.703.2 Field Documentation

8.703.2.1 `uint8_t sMSCAddress::data[256]`

8.703.2.2 `uint8_t sMSCAddress::length`

## 8.704 SMSCAddress Struct Reference

#### Data Fields

- [BYTE length](#)
- [BYTE data](#) [256]

### 8.704.1 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>

### 8.704.2 Field Documentation

8.704.2.1 `BYTE SMSCAddress::data[256]`

8.704.2.2 `BYTE SMSCAddress::length`

## 8.705 sMSCAddressTlv Struct Reference

#### Data Fields

- `uint8_t TlvPresent`
- `sMSCAddressInfo SMSCInfo`

### 8.705.1 Detailed Description

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>SMSCInfo</i>	<ul style="list-style-type: none"> <li>• SMSC Address</li> <li>• See <a href="#">sMSCAddressInfo</a> for more information</li> </ul>

## 8.705.2 Field Documentation

8.705.2.1 `sMSCAddressInfo sMSCAddressTlv::SMSCInfo`8.705.2.2 `uint8_t sMSCAddressTlv::TlvPresent`

## 8.706 sMSEtwsMessage Struct Reference

## Data Fields

- `uint8_t notificationType`
- `uint16_t length`
- `uint8_t data` [1254]

## 8.706.1 Detailed Description

## 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>

## 8.706.2 Field Documentation

8.706.2.1 `uint8_t sMSEtwsMessage::data[1254]`8.706.2.2 `uint16_t sMSEtwsMessage::length`8.706.2.3 `uint8_t sMSEtwsMessage::notificationType`

## 8.707 sMSEtwsMessage Struct Reference

## Data Fields

- `BYTE notificationType`
- `WORD length`
- `BYTE data` [1254]

### 8.707.1 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>

### 8.707.2 Field Documentation

8.707.2.1 **BYTE** SMSEtwsMessage::data[1254]

8.707.2.2 **WORD** SMSEtwsMessage::length

8.707.2.3 **BYTE** SMSEtwsMessage::notificationType

## 8.708 sMSEtwsMessageTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- [sMSEtwsMessageInfo](#) [EtwsMessageInfo](#)

### 8.708.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>EtwsMessage-Info</i>	<ul style="list-style-type: none"> <li>• ETWS Message</li> <li>• See <a href="#">sMSEtwsMessageInfo</a> for more information</li> </ul>

### 8.708.2 Field Documentation

8.708.2.1 **sMSEtwsMessageInfo** sMSEtwsMessageTlv::EtwsMessageInfo

8.708.2.2 **uint8\_t** sMSEtwsMessageTlv::TlvPresent

## 8.709 sMSEtwsPlmn Struct Reference

## Data Fields

- uint16\_t [mobileCountryCode](#)
- uint16\_t [mobileNetworkCode](#)

### 8.709.1 Detailed Description

#### 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>

### 8.709.2 Field Documentation

8.709.2.1 uint16\_t sMSEtwsPlmn::mobileCountryCode

8.709.2.2 uint16\_t sMSEtwsPlmn::mobileNetworkCode

## 8.710 SMSEtwsPlmn Struct Reference

## Data Fields

- [WORD mobileCountryCode](#)
- [WORD mobileNetworkCode](#)

### 8.710.1 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>

### 8.710.2 Field Documentation

8.710.2.1 WORD SMSEtwsPlmn::mobileCountryCode

8.710.2.2 WORD SMSEtwsPlmn::mobileNetworkCode

## 8.711 SMSEventInfo\_s Struct Reference

## Data Fields

- [BYTE smsEventType](#)

- [SMSMTMessageInfo](#) \* [pMTMessageInfo](#)
- [SMSTransferRouteMTMessageInfo](#) \* [pTransferRouteMTMessageInfo](#)
- [SMSMessageModelInfo](#) \* [pMessageModelInfo](#)
- [SMSEtwsMessageInfo](#) \* [pEtwsMessageInfo](#)
- [SMSEtwsPlmnInfo](#) \* [pEtwsPlmnInfo](#)
- [SMSCAddressInfo](#) \* [pSMSCAddressInfo](#)
- [SMSOnIMSInfo](#) \* [pSMSOnIMSInfo](#)

### 8.711.1 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="#">SMSMessageModelInfo</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 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 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 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 NULL, if this event is not present in the smsEventType parameter Note: None</li> </ul>

### 8.711.2 Field Documentation

8.711.2.1 [SMSEtwsMessageInfo](#)\* [SMSEventInfo\\_s::pEtwsMessageInfo](#)

8.711.2.2 [SMSEtwsPlmnInfo](#)\* [SMSEventInfo\\_s::pEtwsPlmnInfo](#)

8.711.2.3 **SMSMessageModelInfo\*** SMSEventInfo\_s::pMessageModelInfo

8.711.2.4 **SMSMTMessageInfo\*** SMSEventInfo\_s::pMTMessageInfo

8.711.2.5 **SMSCAddressInfo\*** SMSEventInfo\_s::pSMSCAddressInfo

8.711.2.6 **SMSOnIMSInfo\*** SMSEventInfo\_s::pSMSOnIMSInfo

8.711.2.7 **SMSTransferRouteMTMessageInfo\*** SMSEventInfo\_s::pTransferRouteMTMessageInfo

8.711.2.8 **BYTE** SMSEventInfo\_s::smsEventType

## 8.712 smsMaxStorageSizeReq Struct Reference

### Data Fields

- [BYTE](#) storageType
- [BYTE](#) \* pMessageMode

### 8.712.1 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>pMessage-Mode(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

### 8.712.2 Field Documentation

8.712.2.1 **BYTE\*** smsMaxStorageSizeReq::pMessageMode

8.712.2.2 **BYTE** smsMaxStorageSizeReq::storageType

## 8.713 smsMaxStorageSizeResp Struct Reference

### Data Fields

- [ULONG](#) maxStorageSize
- [ULONG](#) freeSlots

### 8.713.1 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>

### 8.713.2 Field Documentation

8.713.2.1 **ULONG** smsMaxStorageSizeResp::freeSlots

8.713.2.2 **ULONG** smsMaxStorageSizeResp::maxStorageSize

## 8.714 SMSMemoryInfo Struct Reference

#### Data Fields

- [BYTE](#) storageType
- [BYTE](#) messageMode

### 8.714.1 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>

### 8.714.2 Field Documentation

8.714.2.1 **BYTE** SMSMemoryInfo::messageMode

8.714.2.2 **BYTE** SMSMemoryInfo::storageType

## 8.715 sSMMessageMode Struct Reference

## Data Fields

- uint8\_t [messageMode](#)

### 8.715.1 Detailed Description

#### Parameters

<i>messageMode</i>	Message Mode
--------------------	--------------

### 8.715.2 Field Documentation

#### 8.715.2.1 uint8\_t sSMMessageMode::messageMode

## 8.716 SMSMessageMode Struct Reference

## Data Fields

- BYTE [messageMode](#)

### 8.716.1 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>
--------------------	--

### 8.716.2 Field Documentation

#### 8.716.2.1 BYTE SMSMessageMode::messageMode

## 8.717 smsMsgprotocolResp Struct Reference

## Data Fields

- BYTE [msgProtocol](#)

### 8.717.1 Detailed Description

This structure contains get message protocol response parameters

#### Parameters

<i>msgProtocol</i>	<ul style="list-style-type: none"><li>-</li><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>
--------------------	--

### 8.717.2 Field Documentation

8.717.2.1 **BYTE** smsMsgprotocolResp::msgProtocol

## 8.718 sMSMTMessage Struct Reference

### Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)

### 8.718.1 Detailed Description

#### Parameters

<i>storageType</i>	memory storage 0x00-UIM 0x01-NV
<i>messageIndex</i>	MT Message index

### 8.718.2 Field Documentation

8.718.2.1 **uint32\_t** sMSMTMessage::messageIndex

8.718.2.2 **uint32\_t** sMSMTMessage::storageType

## 8.719 SMSMTMessage Struct Reference

### Data Fields

- [ULONG](#) [storageType](#)
- [ULONG](#) [messageIndex](#)

### 8.719.1 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 0 - UIM 1 - NV</li></ul>
<i>messageIndex</i>	<ul style="list-style-type: none"><li>• Index of the new message</li></ul>

### 8.719.2 Field Documentation

8.719.2.1 **ULONG** SMSMTMessage::messageIndex

8.719.2.2 **ULONG** SMSMTMessage::storageType

## 8.720 sMSOnIMS Struct Reference

### Data Fields

- uint8\_t [smsOnIMS](#)

### 8.720.1 Detailed Description

#### Parameters

<i>smsOnIMS</i>	SMS on IMS
-----------------	------------

### 8.720.2 Field Documentation

#### 8.720.2.1 uint8\_t sMSOnIMS::smsOnIMS

## 8.721 SMSOnIMS Struct Reference

### Data Fields

- [BYTE smsOnIMS](#)

### 8.721.1 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>
-----------------	---

### 8.721.2 Field Documentation

#### 8.721.2.1 BYTE SMSOnIMS::smsOnIMS

## 8.722 sMSOnIMSTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- [sMSOnIMSInfo IMSInfo](#)

### 8.722.1 Detailed Description

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Boolean indicating the presence of the TLV in the QMI response</li> </ul>
<i>IMSInfo</i>	<ul style="list-style-type: none"> <li>• SMS on IMS</li> <li>• See <a href="#">sMSOnIMSInfo</a> for more information</li> </ul>

## 8.722.2 Field Documentation

8.722.2.1 sMSOnIMSInfo sMSOnIMSTlv::IMSInfo

8.722.2.2 uint8\_t sMSOnIMSTlv::TlvPresent

## 8.723 smsRouteEntry Struct Reference

## Data Fields

- [BYTE messageType](#)
- [BYTE messageClass](#)
- [BYTE routeStorage](#)
- [BYTE receiptAction](#)

## 8.723.1 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>
----------------------	---

### 8.723.2 Field Documentation

8.723.2.1 **BYTE** smsRouteEntry::messageClass

8.723.2.2 **BYTE** smsRouteEntry::messageType

8.723.2.3 **BYTE** smsRouteEntry::receiptAction

8.723.2.4 **BYTE** smsRouteEntry::routeStorage

## 8.724 smsSetRoutesReq Struct Reference

### Data Fields

- [WORD](#) numOfRoutes
- [smsRouteEntry](#) routeList [0x0A]
- BYTE** \* [pTransferStatusReport](#)

### 8.724.1 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 report are transferred to the client (optional)</li> </ul>

### 8.724.2 Field Documentation

8.724.2.1 **WORD** smsSetRoutesReq::numOfRoutes

8.724.2.2 **BYTE\*** smsSetRoutesReq::pTransferStatusReport

8.724.2.3 **smsRouteEntry** smsSetRoutesReq::routeList[0x0A]

## 8.725 sMSTransferRouteMTMessage Struct Reference

### Data Fields

- uint8\_t [ackIndicator](#)
- uint32\_t [transactionID](#)
- uint8\_t [format](#)
- uint16\_t [length](#)
- uint8\_t [data](#) [256]

### 8.725.1 Detailed Description

#### 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>

### 8.725.2 Field Documentation

8.725.2.1 uint8\_t sMSTransferRouteMTMessage::ackIndicator

8.725.2.2 uint8\_t sMSTransferRouteMTMessage::data[256]

8.725.2.3 uint8\_t sMSTransferRouteMTMessage::format

8.725.2.4 uint16\_t sMSTransferRouteMTMessage::length

8.725.2.5 uint32\_t sMSTransferRouteMTMessage::transactionID

## 8.726 SMSTransferRouteMTMessage Struct Reference

### Data Fields

- [BYTE](#) [ackIndicator](#)
- [ULONG](#) [transactionID](#)
- [BYTE](#) [format](#)
- [WORD](#) [length](#)
- [BYTE](#) [data](#) [256]

### 8.726.1 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>

### 8.726.2 Field Documentation

8.726.2.1 BYTE SMSTransferRouteMTMessage::ackIndicator

8.726.2.2 BYTE SMSTransferRouteMTMessage::data[256]

8.726.2.3 BYTE SMSTransferRouteMTMessage::format

8.726.2.4 WORD SMSTransferRouteMTMessage::length

8.726.2.5 ULONG SMSTransferRouteMTMessage::transactionID

## 8.727 sQosFlowStat Struct Reference

#### Data Fields

- [ULONG bearerId](#)
- [ULONG tx\\_pkt](#)
- [ULONG tx\\_pkt\\_drp](#)
- [ULONGLONG tx\\_bytes](#)
- [ULONGLONG tx\\_bytes\\_drp](#)

### 8.727.1 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>

## 8.727.2 Field Documentation

8.727.2.1 **ULONG** sQosFlowStat::bearerId

8.727.2.2 **ULONGLONG** sQosFlowStat::tx\_bytes

8.727.2.3 **ULONGLONG** sQosFlowStat::tx\_bytes\_drp

8.727.2.4 **ULONG** sQosFlowStat::tx\_pkt

8.727.2.5 **ULONG** sQosFlowStat::tx\_pkt\_drp

## 8.728 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)]

### 8.728.1 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>

## 8.728.2 Field Documentation

8.728.2.1 **ULONG** sQosStat::apnId

8.728.2.2 **ULONG** sQosStat::numQosFlow

8.728.2.3 **sQosFlowStat** sQosStat::qosFlow[(10)]

8.728.2.4 **ULONGLONG** sQosStat::total\_rx\_bytes

8.728.2.5 **ULONG** sQosStat::total\_rx\_pkt

8.728.2.6 **ULONGLONG** sQosStat::total\_tx\_bytes

8.728.2.7 **ULONGLONG** sQosStat::total\_tx\_bytes\_drp

8.728.2.8 **ULONG** sQosStat::total\_tx\_pkt

8.728.2.9 **ULONG** sQosStat::total\_tx\_pkt\_drp

## 8.729 SrvStatusInfo Struct Reference

### Data Fields

- [BYTE](#) srvStatus
- [BYTE](#) isPrefDataPath

### 8.729.1 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>

### 8.729.2 Field Documentation

#### 8.729.2.1 BYTE SrvStatusInfo::isPrefDataPath

#### 8.729.2.2 BYTE SrvStatusInfo::srvStatus

## 8.730 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](#)

### 8.730.1 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>instanceId</i>	<ul style="list-style-type: none"> <li>• PDP Instance.</li> <li>• Instance ID corresponding to the session ID</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[IN\O-UT]</i>	<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>

## 8.730.2 Field Documentation

8.730.2.1 **BOOL** ssdatasession\_params::action

8.730.2.2 **ULONG** ssdatasession\_params::failureReason

8.730.2.3 **ULONG** ssdatasession\_params::failureReasonv4

8.730.2.4 **ULONG** ssdatasession\_params::failureReasonv6

8.730.2.5 **BYTE** ssdatasession\_params::instanceId

8.730.2.6 **BYTE** ssdatasession\_params::ipfamily

8.730.2.7 **ULONG\*** ssdatasession\_params::pAuthentication

8.730.2.8 **CHAR\*** ssdatasession\_params::pPassword

8.730.2.9 **ULONG\*** ssdatasession\_params::pProfileId3GPP

8.730.2.10 **ULONG\*** ssdatasession\_params::pProfileId3GPP2

- 8.730.2.11 **ULONG\*** ssdatasession\_params::pTechnology
- 8.730.2.12 **CHAR\*** ssdatasession\_params::pUsername
- 8.730.2.13 **ULONG** ssdatasession\_params::rcv4
- 8.730.2.14 **ULONG** ssdatasession\_params::rcv6
- 8.730.2.15 **ULONG** ssdatasession\_params::sessionId
- 8.730.2.16 **ULONG** ssdatasession\_params::v4sessionId
- 8.730.2.17 **ULONG** ssdatasession\_params::v6sessionId
- 8.730.2.18 **ULONG** ssdatasession\_params::verbFailReason
- 8.730.2.19 **ULONG** ssdatasession\_params::verbFailReasonType

## 8.731 SupportedMsgList Struct Reference

### Data Fields

- [WORD](#) supportedMsgLen
- [BYTE](#) supportedMsgs [256]

### 8.731.1 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>

### 8.731.2 Field Documentation

- 8.731.2.1 **WORD** SupportedMsgList::supportedMsgLen
- 8.731.2.2 **BYTE** SupportedMsgList::supportedMsgs[256]

## 8.732 SUPSInfo Struct Reference

### Data Fields

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

### 8.732.1 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>

### 8.732.2 Field Documentation

#### 8.732.2.1 BYTE SUPSInfo::isModByCC

#### 8.732.2.2 BYTE SUPSInfo::svcType

## 8.733 SV Struct Reference

#### Data Fields

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

### 8.733.1 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>

### 8.733.2 Field Documentation

#### 8.733.2.1 WORD SV::id

#### 8.733.2.2 BYTE SV::mask

#### 8.733.2.3 ULONG SV::system

## 8.734 SVInfo Struct Reference

### Data Fields

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

### 8.734.1 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>

### 8.734.2 Field Documentation

8.734.2.1 **BYTE** SVInfo::len8.734.2.2 **SV\*** SVInfo::pSV

## 8.735 svUsedforFix\_s Struct Reference

### Data Fields

- [BYTE](#) gnssSvUsedList\_len
- [WORD](#) gnssSvUsedList [255]

### 8.735.1 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>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>

### 8.735.2 Field Documentation

8.735.2.1 **WORD** svUsedforFix\_s::gnssSvUsedList[255]8.735.2.2 **BYTE** svUsedforFix\_s::gnssSvUsedList\_len

## 8.736 SWI\_STRUCT\_CarrierImage Struct Reference

### Data Fields

- [ULONG](#) m\_nCarrierId
- [ULONG](#) m\_nFolderId
- [ULONG](#) m\_nStorage
- [BYTE](#) m\_FwImageld [16]
- [BYTE](#) m\_FwBuildId [100]
- [BYTE](#) m\_PriImageld [16]
- [BYTE](#) m\_PriBuildId [100]

### 8.736.1 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>

## 8.736.2 Field Documentation

8.736.2.1 BYTE SWI\_STRUCT\_CarrierImage::m\_FwBuildId[100]

8.736.2.2 BYTE SWI\_STRUCT\_CarrierImage::m\_FwImageId[16]

8.736.2.3 ULONG SWI\_STRUCT\_CarrierImage::m\_nCarrierId

8.736.2.4 ULONG SWI\_STRUCT\_CarrierImage::m\_nFolderId

8.736.2.5 ULONG SWI\_STRUCT\_CarrierImage::m\_nStorage

8.736.2.6 BYTE SWI\_STRUCT\_CarrierImage::m\_PriBuildId[100]

8.736.2.7 BYTE SWI\_STRUCT\_CarrierImage::m\_PriImageId[16]

## 8.737 SwiLocGetAutoStartResp Struct Reference

## Data Fields

- BYTE function
- BOOL function\_reported
- BYTE fix\_type
- BOOL fix\_type\_reported
- BYTE max\_time
- BOOL max\_time\_reported
- ULONG max\_dist

- [BOOL max\\_dist\\_reported](#)
- [ULONG fix\\_rate](#)
- [BOOL fix\\_rate\\_reported](#)

### 8.737.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

#### Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>• Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> <li>– 0 - disabled</li> <li>– 1 - At bootup</li> <li>– 2 - When NMEA port is opened</li> </ul> </li> </ul>
<i>function_ - reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>• Type of GNSS fix: <ul style="list-style-type: none"> <li>– 1 - Default Engine mode</li> <li>– 2 - MS-Based</li> <li>– 3 - MS-Assisted</li> <li>– 4 - Standalone</li> </ul> </li> </ul>
<i>fix_type_ - reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>• Maximum time allowed for the receiver to get a fix in seconds</li> <li>• Valid range: 1-255</li> </ul>
<i>max_time_ - reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>max_dist</i>	<ul style="list-style-type: none"> <li>• Maximum uncertainty of a fix measured by distance in meters</li> <li>• Valid range: 1 - 4294967280</li> </ul>
<i>max_dist_ - reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>• Time between fixes in seconds</li> <li>• Valid range: 1–65535</li> </ul>

<i>fix_rate_reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
--------------------------	--

## 8.737.2 Field Documentation

8.737.2.1 **ULONG** SwiLocGetAutoStartResp::fix\_rate

8.737.2.2 **BOOL** SwiLocGetAutoStartResp::fix\_rate\_reported

8.737.2.3 **BYTE** SwiLocGetAutoStartResp::fix\_type

8.737.2.4 **BOOL** SwiLocGetAutoStartResp::fix\_type\_reported

8.737.2.5 **BYTE** SwiLocGetAutoStartResp::function

8.737.2.6 **BOOL** SwiLocGetAutoStartResp::function\_reported

8.737.2.7 **ULONG** SwiLocGetAutoStartResp::max\_dist

8.737.2.8 **BOOL** SwiLocGetAutoStartResp::max\_dist\_reported

8.737.2.9 **BYTE** SwiLocGetAutoStartResp::max\_time

8.737.2.10 **BOOL** SwiLocGetAutoStartResp::max\_time\_reported

## 8.738 SwiLocSetAutoStartReq Struct Reference

### Data Fields

- [BYTE function](#)
- [BOOL set\\_function](#)
- [BYTE fix\\_type](#)
- [BOOL set\\_fix\\_type](#)
- [BYTE max\\_time](#)
- [BOOL set\\_max\\_time](#)
- [ULONG max\\_dist](#)
- [BOOL set\\_max\\_dist](#)
- [ULONG fix\\_rate](#)
- [BOOL set\\_fix\\_rate](#)

### 8.738.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

#### Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>• Setting to indicate when modem should start an automatic GNSS fix             <ul style="list-style-type: none"> <li>– 0 - disabled</li> <li>– 1 - At bootup</li> <li>– 2 - When NMEA port is opened</li> </ul> </li> </ul>
-----------------	--

<i>set_function</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>• Type of GNSS fix: <ul style="list-style-type: none"> <li>– 1 - Default Engine mode</li> <li>– 2 - MS-Based</li> <li>– 3 - MS-Assisted</li> <li>– 4 - Standalone</li> </ul> </li> </ul>
<i>set_fix_type</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>• Maximum time allowed for the receiver to get a fix in seconds</li> <li>• Valid range: 1-255</li> </ul>
<i>set_max_time</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>max_dist</i>	<ul style="list-style-type: none"> <li>• Maximum uncertainty of a fix measured by distance in meters</li> <li>• Valid range: 1 - 4294967280</li> </ul>
<i>set_max_dist</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>• Time between fixes in seconds</li> <li>• Valid range: 1–65535</li> </ul>
<i>set_fix_rate</i>	<ul style="list-style-type: none"> <li>• 0 - do not set to modem</li> <li>• 1 - set to modem</li> </ul>

## 8.738.2 Field Documentation

8.738.2.1 **ULONG** SwiLocSetAutoStartReq::fix\_rate

8.738.2.2 **BYTE** SwiLocSetAutoStartReq::fix\_type

8.738.2.3 **BYTE** SwiLocSetAutoStartReq::function

8.738.2.4 **ULONG** SwiLocSetAutoStartReq::max\_dist

8.738.2.5 **BYTE** SwiLocSetAutoStartReq::max\_time

8.738.2.6 **BOOL** SwiLocSetAutoStartReq::set\_fix\_rate

8.738.2.7 **BOOL** SwiLocSetAutoStartReq::set\_fix\_type

8.738.2.8 **BOOL** SwiLocSetAutoStartReq::set\_function

8.738.2.9 **BOOL** SwiLocSetAutoStartReq::set\_max\_dist

8.738.2.10 **BOOL** SwiLocSetAutoStartReq::set\_max\_time

## 8.739 swiModemStatusResp Struct Reference

### Data Fields

- [CommInfo](#) [commonInfo](#)
- [LTEInfo](#) \* [pLTEInfo](#)

### 8.739.1 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>

### 8.739.2 Field Documentation

8.739.2.1 **CommInfo** swiModemStatusResp::commonInfo

8.739.2.2 **LTEInfo**\* swiModemStatusResp::pLTEInfo

## 8.740 SwiOTAMsg\_s Struct Reference

### Data Fields

- [ULONG](#) type
- [WORD](#) data\_len
- [BYTE](#) data [2048]
- [LteNasReleaseInfo](#) \* [pLteNasRelInfo](#)
- [ULONGLONG](#) \* [pTime](#)

### 8.740.1 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>

## 8.740.2 Field Documentation

8.740.2.1 BYTE SwiOTAMsg\_s::data[2048]

8.740.2.2 WORD SwiOTAMsg\_s::data\_len

8.740.2.3 LteNasReleaseInfo\* SwiOTAMsg\_s::pLteNasRelInfo

8.740.2.4 ULONGLONG\* SwiOTAMsg\_s::pTime

8.740.2.5 ULONG SwiOTAMsg\_s::type

## 8.741 swiPDPRuntimeSettingsReq Struct Reference

## Data Fields

- [BYTE contextId](#)
- [BYTE contextType](#)

## 8.741.1 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>

## 8.741.2 Field Documentation

8.741.2.1 **BYTE** swiPDPRuntimeSettingsReq::contextId

8.741.2.2 **BYTE** swiPDPRuntimeSettingsReq::contextType

## 8.742 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

### 8.742.1 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.

**8.742.2 Field Documentation**

- 8.742.2.1 **CHAR\*** swiPDPRuntimeSettingsResp::pAPNName
- 8.742.2.2 **BYTE\*** swiPDPRuntimeSettingsResp::pBearerId
- 8.742.2.3 **BYTE\*** swiPDPRuntimeSettingsResp::pContextId
- 8.742.2.4 **ULONG\*** swiPDPRuntimeSettingsResp::pIPv4Address
- 8.742.2.5 **ULONG\*** swiPDPRuntimeSettingsResp::pIPv4GWAddress
- 8.742.2.6 **struct IPV6AddressInfo\*** swiPDPRuntimeSettingsResp::pIPv6Address
- 8.742.2.7 **struct IPV6AddressInfo\*** swiPDPRuntimeSettingsResp::pIPv6GWAddress
- 8.742.2.8 **ULONG\*** swiPDPRuntimeSettingsResp::pPrDNSIPv4Address
- 8.742.2.9 **WORD\*** swiPDPRuntimeSettingsResp::pPrDNSIPv6Address
- 8.742.2.10 **ULONG\*** swiPDPRuntimeSettingsResp::pPrPCSCFIPv4Address
- 8.742.2.11 **WORD\*** swiPDPRuntimeSettingsResp::pPrPCSCFIPv6Address
- 8.742.2.12 **ULONG\*** swiPDPRuntimeSettingsResp::pSeDNSIPv4Address
- 8.742.2.13 **WORD\*** swiPDPRuntimeSettingsResp::pSeDNSIPv6Address
- 8.742.2.14 **ULONG\*** swiPDPRuntimeSettingsResp::pSePCSCFIPv4Address
- 8.742.2.15 **WORD\*** swiPDPRuntimeSettingsResp::pSePCSCFIPv6Address

**8.743 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](#) \* [pId](#)
- [Port](#) \* [pTranSrcPort](#)
- [Port](#) \* [pTranDstPort](#)

### 8.743.1 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>pld</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>

## 8.743.2 Field Documentation

8.743.2.1 **BYTE** swiQosFilter::index

8.743.2.2 **ULONG\*** swiQosFilter::pEspSpi

8.743.2.3 **WORD\*** swiQosFilter::pld

8.743.2.4 **IPv4Addr\*** swiQosFilter::pIPv4DstAddr

8.743.2.5 **IPv4Addr\*** swiQosFilter::pIPv4SrcAddr

8.743.2.6 **IPv6Addr\*** swiQosFilter::pIPv6DstAddr

8.743.2.7 **ULONG\*** swiQosFilter::pIPv6Label

8.743.2.8 **IPv6Addr\*** swiQosFilter::pIPv6SrcAddr

8.743.2.9 **IPv6TrafCls\*** swiQosFilter::pIPv6TrafCls

8.743.2.10 **BYTE\*** swiQosFilter::pNextHdrProto

8.743.2.11 **WORD\*** swiQosFilter::pPrecedence

8.743.2.12 **Port\*** swiQosFilter::pTCPDstPort

8.743.2.13 **Port\*** swiQosFilter::pTCPSrcPort

8.743.2.14 **Tos\*** swiQosFilter::pTos

8.743.2.15 **Port\*** swiQosFilter::pTranDstPort

8.743.2.16 **Port\*** swiQosFilter::pTranSrcPort

8.743.2.17 **Port\*** swiQosFilter::pUDPDstPort

8.743.2.18 Port\* swiQosFilter::pUDPSrcPort

8.743.2.19 BYTE swiQosFilter::version

## 8.744 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

### 8.744.1 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: <ul style="list-style-type: none"> <li>• 0 – Conversational</li> <li>• 1 – Streaming</li> <li>• 2 – Interactive</li> <li>• 3 – Background</li> </ul> </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>pMinPolicedPktSz</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>pMaxAllowedPktSz</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>p3GPPResResidualBER</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>

## 8.744.2 Field Documentation

8.744.2.1 **BYTE** swiQosFlow::index

8.744.2.2 **BYTE\*** swiQosFlow::p3GPP2Pri

8.744.2.3 **BYTE\*** swiQosFlow::p3GPPImCn

8.744.2.4 **WORD\*** swiQosFlow::p3GPPResResidualBER

8.744.2.5 **BYTE\*** swiQosFlow::p3GPPSigInd

8.744.2.6 **BYTE\*** swiQosFlow::p3GPPTraHdlPri

8.744.2.7 **dataRate\*** swiQosFlow::pDataRate

8.744.2.8 **ULONG\*** swiQosFlow::pJitter

8.744.2.9 **ULONG\*** swiQosFlow::pLatency

8.744.2.10 **BYTE\*** swiQosFlow::pLteQci

8.744.2.11 **ULONG\*** `swiQosFlow::pMaxAllowedPktSz`

8.744.2.12 **ULONG\*** `swiQosFlow::pMinPolicedPktSz`

8.744.2.13 **pktErrRate\*** `swiQosFlow::pPktErrRate`

8.744.2.14 **WORD\*** `swiQosFlow::pProfileId3GPP2`

8.744.2.15 **tokenBucket\*** `swiQosFlow::pTokenBucket`

8.744.2.16 **BYTE\*** `swiQosFlow::pTrafficClass`

## 8.745 swiQosGranted Struct Reference

### Data Fields

- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)

### 8.745.1 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

### 8.745.2 Field Documentation

8.745.2.1 **swiQosFlow\*** `swiQosGranted::pRxFlow`

8.745.2.2 **swiQosFlow\*** `swiQosGranted::pTxFlow`

## 8.746 swiQosIds Struct Reference

### Data Fields

- [BYTE](#) `sz`
- [ULONG](#) \* `plds`

### 8.746.1 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

### 8.746.2 Field Documentation

8.746.2.1 **ULONG\*** swiQosIds::pIds

8.746.2.2 **BYTE** swiQosIds::sz

## 8.747 swiQosModifyReq Struct Reference

### Data Fields

- [ULONG](#) id
- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)
- [swiQosFilter](#) \* [pTxFilter](#)
- [swiQosFilter](#) \* [pRxFilter](#)

### 8.747.1 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_S_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

### 8.747.2 Field Documentation

8.747.2.1 **ULONG** swiQosModifyReq::id

8.747.2.2 **swiQosFilter\*** swiQosModifyReq::pRxFilter

8.747.2.3 **swiQosFlow\*** swiQosModifyReq::pRxFlow

8.747.2.4 **swiQosFilter\*** swiQosModifyReq::pTxFilter

8.747.2.5 **swiQosFlow\*** swiQosModifyReq::pTxFlow

## 8.748 swiQosReq Struct Reference

### Data Fields

- [BYTE](#) index
- [swiQosFlow](#) \* [pTxFlow](#)
- [swiQosFlow](#) \* [pRxFlow](#)
- [swiQosFilter](#) \* [pTxFilter](#)
- [swiQosFilter](#) \* [pRxFilter](#)

### 8.748.1 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>

## 8.748.2 Field Documentation

8.748.2.1 BYTE swiQosReq::index

8.748.2.2 swiQosFilter\* swiQosReq::pRxFilter

8.748.2.3 swiQosFlow\* swiQosReq::pRxFlow

8.748.2.4 swiQosFilter\* swiQosReq::pTxFilter

8.748.2.5 swiQosFlow\* swiQosReq::pTxFlow

## 8.749 swiRMTrasferStaticsReq Struct Reference

## Data Fields

- [BYTE bResetStatistics](#)
- [ULONG ulMask](#)

## 8.749.1 Detailed Description

RM Transfer Satistics 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 Indication 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>
---------------	--

## 8.749.2 Field Documentation

8.749.2.1 **BYTE** `swiRMTransferStaticsReq::bResetStatistics`

8.749.2.2 **ULONG** `swiRMTransferStaticsReq::ulMask`

## 8.750 sysInfoCommon Struct Reference

### Data Fields

- [BYTE](#) `srvDomainValid`
- [BYTE](#) `srvDomain`
- [BYTE](#) `srvCapabilityValid`
- [BYTE](#) `srvCapability`
- [BYTE](#) `roamStatusValid`
- [BYTE](#) `roamStatus`
- [BYTE](#) `isSysForbiddenValid`
- [BYTE](#) `isSysForbidden`

### 8.750.1 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> <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 Numbers</li> <li>0x40 to 0xFF - Reserved.</li> <li>0xFF - Not Available</li> </ul> </li> <li>Values from 0x02 onward are only applicable for 3GPP2</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>
-----------------------	--

## 8.750.2 Field Documentation

8.750.2.1 **BYTE** sysInfoCommon::isSysForbidden

8.750.2.2 **BYTE** sysInfoCommon::isSysForbiddenValid

8.750.2.3 **BYTE** sysInfoCommon::roamStatus

8.750.2.4 **BYTE** sysInfoCommon::roamStatusValid

8.750.2.5 **BYTE** sysInfoCommon::srvCapability

8.750.2.6 **BYTE** sysInfoCommon::srvCapabilityValid

8.750.2.7 **BYTE** sysInfoCommon::srvDomain

8.750.2.8 **BYTE** sysInfoCommon::srvDomainValid

## 8.751 t\_gpsTime Struct Reference

### Data Fields

- [USHORT](#) gpsWeek
- [ULONG](#) gpsTimeOfWeekMs

## 8.751.1 Field Documentation

8.751.1.1 **ULONG** t\_gpsTime::gpsTimeOfWeekMs

8.751.1.2 **USHORT** t\_gpsTime::gpsWeek

## 8.752 t\_sensor Struct Reference

### Data Fields

- [ULONG](#) usageMask
- [ULONG](#) aidingIndicatorMask

## 8.752.1 Field Documentation

8.752.1.1 **ULONG** t\_sensor::aidingIndicatorMask

8.752.1.2 **ULONG** t\_sensor::usageMask

## 8.753 t\_Sv Struct Reference

### Data Fields

- [BYTE len](#)
- [USHORT entries](#) [255]

### 8.753.1 Field Documentation

#### 8.753.1.1 USHORT t\_Sv::entries[255]

#### 8.753.1.2 BYTE t\_Sv::len

## 8.754 TDSCDMAECIOThresh Struct Reference

### Data Fields

- [BYTE TDSCDMAECIOThreshListLen](#)
- [ULONG \\* pTDSCDMAECIOThreshList](#)

### 8.754.1 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>

### 8.754.2 Field Documentation

#### 8.754.2.1 ULONG\* TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

#### 8.754.2.2 BYTE TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.755 TDSCDMARSCPThresh Struct Reference

### Data Fields

- [BYTE TDSCDMARSCPThreshListLen](#)
- [WORD \\* pTDSCDMARSCPThreshList](#)

### 8.755.1 Detailed Description

This structure contains TDSCDMA RSCP threshold related parameters.

## Parameters

<i>TDSCDMARSC- PThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the TDSCDMA RSCP threshold list parameter to follow</li> </ul>
<i>pTDSCDMARSC- CThreshList</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>

## 8.755.2 Field Documentation

8.755.2.1 WORD\* TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.755.2.2 BYTE TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.756 TDSCDMARSSIThresh Struct Reference

## Data Fields

- [BYTE TDSCDMARSSIThreshListLen](#)
- [ULONG \\* pTDSCDMARSSIThreshList](#)

## 8.756.1 Detailed Description

This structure contains TDSCDMA RSSI threshold related parameters.

## Parameters

<i>TDSCDMARSS- IThreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the TDSCDMA RSSI threshold list parameter to follow</li> </ul>
<i>pTDSCDMARSS- SIThreshList</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>

## 8.756.2 Field Documentation

8.756.2.1 ULONG\* TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.756.2.2 BYTE TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.757 TDSCDMASigInfoExt Struct Reference

## Data Fields

- [FLOAT rssi](#)
- [FLOAT rscp](#)
- [FLOAT ecio](#)
- [FLOAT sinr](#)

### 8.757.1 Detailed Description

This structure contains the TDSCDMA Signal Strength Info Extended

#### Parameters

<i>rss</i>	<ul style="list-style-type: none"> <li>Measured RSSI in dB</li> </ul>
<i>rscp</i> [Optional]	<ul style="list-style-type: none"> <li>Measured RSCP in dBm</li> </ul>
<i>ecio</i> [Optional]	<ul style="list-style-type: none"> <li>Measured ECIO in dBm.</li> </ul>
<i>sinr</i> [Optional]	<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>

### 8.757.2 Field Documentation

8.757.2.1 FLOAT TDSCDMASigInfoExt::ecio

8.757.2.2 FLOAT TDSCDMASigInfoExt::rscp

8.757.2.3 FLOAT TDSCDMASigInfoExt::rss

8.757.2.4 FLOAT TDSCDMASigInfoExt::sinr

## 8.758 tdscdmaSigInfoExt Struct Reference

### Data Fields

- float [rss](#)
- float [rscp](#)
- float [ecio](#)
- float [sinr](#)

### 8.758.1 Detailed Description

#### Parameters

<i>rss</i>	RSSI in dBm.
<i>rsrq</i>	RSRQ value in dB
<i>rsrp</i>	Current RSRP in dBm as measured by L1.
<i>snr</i>	SNR level as a scaled integer in units of 0.1 dB.

### 8.758.2 Field Documentation

8.758.2.1 float tdscdmaSigInfoExt::ecio

8.758.2.2 float tdscdmaSigInfoExt::rscp

8.758.2.3 float tdscdmaSigInfoExt::rsi

8.758.2.4 float tdscdmaSigInfoExt::sinr

## 8.759 TDSCDMASINRCONFThresh Struct Reference

### Data Fields

- [BYTE TDSCDMASINRCONFThreshListLen](#)
- [FLOAT \\* pTDSCDMASINRCONFThreshList](#)

### 8.759.1 Detailed Description

This structure contains TDSCDMA SINR threshold related parameters.

#### Parameters

<i>TDSCDMASIN- RCONFThresh- ListLen</i>	<ul style="list-style-type: none"> <li>• Length of the TDSCDMA SINR threshold list parameter to follow</li> </ul>
<i>pTDSCDMASIN- RCONFThresh- List</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>

### 8.759.2 Field Documentation

8.759.2.1 [FLOAT\\*](#) TDSCDMASINRCONFThresh::pTDSCDMASINRCONFThreshList

8.759.2.2 [BYTE](#) TDSCDMASINRCONFThresh::TDSCDMASINRCONFThreshListLen

## 8.760 TDSCDMASINRThresh Struct Reference

### Data Fields

- [BYTE TDSCDMASINRThreshListLen](#)
- [ULONG \\* pTDSCDMASINRThreshList](#)

### 8.760.1 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>

## 8.760.2 Field Documentation

8.760.2.1 **ULONG\*** TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.760.2.2 **BYTE** TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.761 tempratureData Struct Reference

### Data Fields

- [ULONG](#) timeSource
- [ULONG](#) timeOfFirstSample
- [BYTE](#) temperatureDataLen
- [WORD](#) timeOffset [64]
- [ULONG](#) temperature [64]

### 8.761.1 Detailed Description

This structure specifies information regarding the Temperature Data.

#### Parameters

<i>timeSource</i>	<ul style="list-style-type: none"> <li>• Time source of the sensor data</li> <li>• Valid values <ul style="list-style-type: none"> <li>– 0 - Sensor time source is unspecified</li> <li>– 1 - Time source is common between the sensors and the location engine</li> </ul> </li> </ul>
<i>timeOfFirstSample</i>	<ul style="list-style-type: none"> <li>• Denotes a full 32-bit time stamp of the first (oldest) sample in this message.</li> <li>• The time stamp is in the time reference scale that is used by the sensor time source.</li> <li>• Units - Milliseconds</li> </ul>
<i>temperatureDataLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– timeOffset</li> <li>– temperature</li> </ul> </li> </ul>
<i>timeOffset</i>	<ul style="list-style-type: none"> <li>• Sample time offset</li> <li>• Units - Milliseconds</li> </ul>
<i>temperature</i>	<ul style="list-style-type: none"> <li>• Sensor temperature.</li> <li>• Type - Floating point</li> <li>• Units - Degrees Celsius</li> <li>• Range -50 to +100.00</li> </ul>

### 8.761.2 Field Documentation

8.761.2.1 **ULONG** tempratureData::temperature[64]

8.761.2.2 **BYTE** tempratureData::temperatureDataLen

8.761.2.3 **ULONG** tempratureData::timeOfFirstSample

8.761.2.4 **WORD** tempratureData::timeOffset[64]

8.761.2.5 **ULONG** tempratureData::timeSource

## 8.762 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](#)

### 8.762.1 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>

## 8.762.2 Field Documentation

8.762.2.1 WORD TFTIDParams::destPortRangeEnd

8.762.2.2 WORD TFTIDParams::destPortRangeStart

8.762.2.3 BYTE TFTIDParams::eValid

8.762.2.4 BYTE TFTIDParams::filterId

8.762.2.5 ULONG TFTIDParams::flowLabel

8.762.2.6 ULONG TFTIDParams::IPSECSPi

8.762.2.7 BYTE TFTIDParams::ipVersion

8.762.2.8 BYTE TFTIDParams::nextHeader

8.762.2.9 WORD\* TFTIDParams::pSourceIP

8.762.2.10 BYTE TFTIDParams::sourceIPMask

8.762.2.11 WORD TFTIDParams::srcPortRangeEnd

8.762.2.12 WORD TFTIDParams::srcPortRangeStart

8.762.2.13 WORD TFTIDParams::tosMask

## 8.763 timeInfo Struct Reference

### Data Fields

- [WORD year](#)
- [BYTE month](#)
- [BYTE day](#)
- [BYTE hour](#)
- [BYTE minute](#)
- [BYTE second](#)
- [BYTE dayOfWeek](#)
- [INT8 timeZone](#)
- [BYTE dayLtSavingAdj](#)
- [BYTE radioInterface](#)
- [BYTE TlvPresent](#)

### 8.763.1 Detailed Description

This structure contains the parameters for Network Time.

#### 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</li> <li>• 1 is January and 12 is December</li> </ul>
<i>day</i>	<ul style="list-style-type: none"> <li>• Day</li> <li>• Range - 1 to 31</li> </ul>
<i>hour</i>	<ul style="list-style-type: none"> <li>• Hour</li> <li>• Range - 0 to 59</li> </ul>
<i>minute</i>	<ul style="list-style-type: none"> <li>• Minute</li> <li>• Range - 0 to 59</li> </ul>
<i>second</i>	<ul style="list-style-type: none"> <li>• Second</li> <li>• Range - 0 to 59</li> </ul>
<i>dayOfWeek</i>	<ul style="list-style-type: none"> <li>• Day of the week</li> <li>• 0 is Monday and 6 is Sunday</li> </ul>
<i>timeZone</i>	<ul style="list-style-type: none"> <li>• Offset from Universal time</li> <li>• The difference between local time and Universal time, in increments of 15 min</li> <li>• Signed Value</li> </ul>

<i>dayLtSavingAdj</i>	<ul style="list-style-type: none"> <li>• Daylight saving adjustment in hours</li> <li>• Possible values - 0, 1, and 2.</li> <li>• This field is ignored if radio_if is NAS_RADIO_IF_CDMA_1XEVD0</li> </ul>
<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface from which the information comes</li> <li>• 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>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.763.2 Field Documentation

8.763.2.1 **BYTE** `timeInfo::day`

8.763.2.2 **BYTE** `timeInfo::dayLtSavingAdj`

8.763.2.3 **BYTE** `timeInfo::dayOfWeek`

8.763.2.4 **BYTE** `timeInfo::hour`

8.763.2.5 **BYTE** `timeInfo::minute`

8.763.2.6 **BYTE** `timeInfo::month`

8.763.2.7 **BYTE** `timeInfo::radioInterface`

8.763.2.8 **BYTE** `timeInfo::second`

8.763.2.9 **INT8** `timeInfo::timeZone`

8.763.2.10 **BYTE** `timeInfo::TlvPresent`

8.763.2.11 **WORD** `timeInfo::year`

## 8.764 TmdDeRegNotMitigationLvlReq Struct Reference

### Data Fields

- [BYTE](#) `mitigationDevIDLen`
- [CHAR](#) `mitigationDevID` [255]

### 8.764.1 Detailed Description

This structure contains mitigation devices Level deregister request parameters

#### Parameters

<i>mitigationDevIDLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements <ul style="list-style-type: none"> <li>mitigation_dev_id</li> </ul> </li> </ul>
<i>mitigationDevID</i>	<ul style="list-style-type: none"> <li>Mitigation device ID</li> </ul>

### 8.764.2 Field Documentation

8.764.2.1 CHAR TmdDeRegNotMitigationLvlReq::mitigationDevID[255]

8.764.2.2 BYTE TmdDeRegNotMitigationLvlReq::mitigationDevIDLen

## 8.765 TmdGetMitigationDevListResp Struct Reference

#### Data Fields

- [BYTE \\* pMitigationDevListLen](#)
- [mitigationDevList \\* pMitigationDevList](#)

### 8.765.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

#### Parameters

<i>pMitigationDevListLen</i>	<ul style="list-style-type: none"> <li>Mitigation Device List Length (Optional)</li> <li>Number of sets of the following elements</li> <li>pMitigationDevList</li> </ul>
<i>pMitigationDevList</i>	<ul style="list-style-type: none"> <li>Mitigation Device List (Optional)</li> <li>See <a href="#">mitigationDevList</a> for more information.</li> </ul>

### 8.765.2 Field Documentation

8.765.2.1 mitigationDevList\* TmdGetMitigationDevListResp::pMitigationDevList

8.765.2.2 BYTE\* TmdGetMitigationDevListResp::pMitigationDevListLen

## 8.766 TmdGetMitigationLvlReq Struct Reference

## Data Fields

- [BYTE mitigationDevIDLen](#)
- [CHAR mitigationDevID](#) [255]

### 8.766.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>mitigationDevIDLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– mitigation_dev_id</li> </ul> </li> </ul>
<i>mitigationDevID</i>	<ul style="list-style-type: none"> <li>• Mitigation device ID</li> </ul>

### 8.766.2 Field Documentation

8.766.2.1 **CHAR** TmdGetMitigationLvlReq::mitigationDevID[255]

8.766.2.2 **BYTE** TmdGetMitigationLvlReq::mitigationDevIDLen

## 8.767 TmdGetMitigationLvlResp Struct Reference

## Data Fields

- [BYTE \\* pCurrentmitigationLvl](#)
- [BYTE \\* pReqMitigationLvl](#)

### 8.767.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>pCurrentmitigationLvl</i>	<ul style="list-style-type: none"> <li>• Current thermal mitigation level (Optional)</li> </ul>
<i>pReqMitigationLvl</i>	<ul style="list-style-type: none"> <li>• Requested Thermal Mitigation Level (Optional)</li> <li>• The requested thermal mitigation level from the client. The default is zero if the client has not previously set the mitigation level.</li> </ul>

### 8.767.2 Field Documentation

8.767.2.1 **BYTE\*** TmdGetMitigationLvlResp::pCurrentmitigationLvl

8.767.2.2 **BYTE\*** TmdGetMitigationLvlResp::pReqMitigationLvl

## 8.768 TmdMitigationLvlIndReq Struct Reference

### Data Fields

- [BYTE mitigationDevIDLen](#)
- [CHAR mitigationDevID](#) [255]

### 8.768.1 Detailed Description

This structure contains mitigation Level Indication request parameters

#### Parameters

<i>mitigationDevIDLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements<ul style="list-style-type: none"><li>– mitigation_dev_id</li></ul></li></ul>
<i>mitigationDevID</i>	<ul style="list-style-type: none"><li>• Mitigation device ID</li></ul>

### 8.768.2 Field Documentation

8.768.2.1 **CHAR** TmdMitigationLvlIndReq::mitigationDevID[255]

8.768.2.2 **BYTE** TmdMitigationLvlIndReq::mitigationDevIDLen

## 8.769 TmdRegNotMitigationLvlReq Struct Reference

### Data Fields

- [BYTE mitigationDevIDLen](#)
- [CHAR mitigationDevID](#) [255]

### 8.769.1 Detailed Description

This structure contains mitigation devices Level register request parameters

#### Parameters

<i>mitigationDevIDLen</i>	<ul style="list-style-type: none"><li>• Number of sets of the following elements<ul style="list-style-type: none"><li>– mitigation_dev_id</li></ul></li></ul>
<i>mitigationDevID</i>	<ul style="list-style-type: none"><li>• Mitigation device ID</li></ul>

### 8.769.2 Field Documentation

8.769.2.1 **CHAR** TmdRegNotMitigationLvlReq::mitigationDevID[255]

8.769.2.2 **BYTE** TmdRegNotMitigationLvIReq::mitigationDevIDLen

## 8.770 tokenBucket Struct Reference

### Data Fields

- [ULONG](#) peakRate
- [ULONG](#) tokenRate
- [ULONG](#) bucketSz

### 8.770.1 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

### 8.770.2 Field Documentation

8.770.2.1 **ULONG** tokenBucket::bucketSz

8.770.2.2 **ULONG** tokenBucket::peakRate

8.770.2.3 **ULONG** tokenBucket::tokenRate

## 8.771 Tos Struct Reference

### Data Fields

- [BYTE](#) val
- [BYTE](#) mask

### 8.771.1 Detailed Description

This structure contains the IPv4 filter type of service

#### Parameters

<i>val</i>	Type of service value
<i>mask</i>	<p>Packet matches the TOS filter if: (IPv4_filter_tos_val and IPv4_filter_tos_mask) == (TOS value in the IP packet &amp; IPv4_filter_tos_mask) Example:</p> <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>

## 8.771.2 Field Documentation

8.771.2.1 `BYTE` `Tos::mask`

8.771.2.2 `BYTE` `Tos::val`

## 8.772 transferRouteMessageTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- [sMSTransferRouteMTMessageInfo](#) `TransferRouteMTMessageInfo`

### 8.772.1 Detailed Description

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Boolean indicating the presence of the TLV in the QMI response</li></ul>
<i>TransferRouteMTMessageInfo</i>	<ul style="list-style-type: none"><li>• Transfer Route MT Message</li><li>• See <a href="#">sMSTransferRouteMTMessageInfo</a> for more information</li></ul>

## 8.772.2 Field Documentation

8.772.2.1 `uint8_t` `transferRouteMessageTlv::TlvPresent`

8.772.2.2 `sMSTransferRouteMTMessageInfo` `transferRouteMessageTlv::TransferRouteMTMessageInfo`

## 8.773 TransferStatInd Struct Reference

### Data Fields

- `BYTE` `StatsPeriod`
- `ULONG` `StatsMask`

### 8.773.1 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>
------------------	---

### 8.773.2 Field Documentation

#### 8.773.2.1 ULONG TransferStatInd::StatsMask

#### 8.773.2.2 BYTE TransferStatInd::StatsPeriod

## 8.774 transferStatInd Struct Reference

### Data Fields

- uint8\_t [StatsPeriod](#)
- uint32\_t [StatsMask](#)

### 8.774.1 Detailed Description

#### Parameters

<i>StatsPeriod</i>	Field Period between transfer statistic reports.
<i>StatsMask</i>	requested statistic bit mask.

### 8.774.2 Field Documentation

#### 8.774.2.1 uint32\_t transferStatInd::StatsMask

#### 8.774.2.2 uint8\_t transferStatInd::StatsPeriod

## 8.775 TransferStatsDataType Struct Reference

### Data Fields

- [BYTE interval](#)

### 8.775.1 Field Documentation

#### 8.775.1.1 BYTE TransferStatsDataType::interval

## 8.776 TrStatInd Struct Reference

### Data Fields

- [BYTE statsPeriod](#)
- [ULONG statsMask](#)

### 8.776.1 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 SLQSWdsEventReport-Callback.</li> <li>All unlisted bits are reserved for future use and must be set to zero.</li> </ul>

### 8.776.2 Field Documentation

8.776.2.1 **ULONG** TrStatInd::statsMask

8.776.2.2 **BYTE** TrStatInd::statsPeriod

## 8.777 trueIMSI Struct Reference

### Data Fields

- [BYTE](#) mccT [3]
- [WORD](#) imsiT1112
- [BYTE](#) imsiTS1 [7]
- [BYTE](#) imsiTS2 [3]
- [BYTE](#) imsiTaddrNum

### 8.777.1 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>

## 8.777.2 Field Documentation

8.777.2.1 **WORD** trueIMSI::imsiT1112

8.777.2.2 **BYTE** trueIMSI::imsiTaddrNum

8.777.2.3 **BYTE** trueIMSI::imsiTS1[7]

8.777.2.4 **BYTE** trueIMSI::imsiTS2[3]

8.777.2.5 **BYTE** trueIMSI::mccT[3]

## 8.778 TXAGCList Struct Reference

### Data Fields

- **WORD** \* pTXStaticGain
- **WORD** \* pTXAIG
- **WORD** \* pTXExpThres
- **WORD** \* pTXExpSlope
- **WORD** \* pTXComprThres
- **WORD** \* pTXComprSlope

### 8.778.1 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>

## 8.778.2 Field Documentation

8.778.2.1 WORD\* TXAGCList::pTXAIG

8.778.2.2 WORD\* TXAGCList::pTXComprSlope

8.778.2.3 WORD\* TXAGCList::pTXComprThres

8.778.2.4 WORD\* TXAGCList::pTXExpSlope

8.778.2.5 WORD\* TXAGCList::pTXExpThres

8.778.2.6 WORD\* TXAGCList::pTXStaticGain

## 8.779 txInfo Struct Reference

### Data Fields

- [BYTE isInTraffic](#)
- [INT32 txPower](#)

## 8.779.1 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. <ul style="list-style-type: none"> <li>0x00 - not in traffic</li> <li>0x01 - in traffic</li> </ul> </li> <li>The txPower field is only meaningful when in the device is in traffic.</li> </ul>
<i>txPower</i>	<ul style="list-style-type: none"> <li>Tx power value in 1/10 dbm.</li> </ul>

## 8.779.2 Field Documentation

8.779.2.1 BYTE txInfo::isInTraffic

## 8.779.2.2 INT32 txInfo::txPower

## 8.780 TXPCMIIRFiltr Struct Reference

## Data Fields

- WORD \* pFlag
- WORD \* pStageCnt
- BYTE \* pStage0Val
- BYTE \* pStage1Val
- BYTE \* pStage2Val
- BYTE \* pStage3Val
- BYTE \* pStage4Val

## 8.780.1 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>

## 8.780.2 Field Documentation

8.780.2.1 WORD\* TXPCMIIRFitr::pFlag

8.780.2.2 BYTE\* TXPCMIIRFitr::pStage0Val

8.780.2.3 BYTE\* TXPCMIIRFitr::pStage1Val

8.780.2.4 BYTE\* TXPCMIIRFitr::pStage2Val

8.780.2.5 BYTE\* TXPCMIIRFitr::pStage3Val

8.780.2.6 BYTE\* TXPCMIIRFitr::pStage4Val

8.780.2.7 WORD\* TXPCMIIRFitr::pStageCnt

## 8.781 uim\_appStatus Struct Reference

### Data Fields

- uint8\_t [appType](#)
- uint8\_t [appState](#)
- uint8\_t [persoState](#)
- uint8\_t [persoFeature](#)
- uint8\_t [persoRetries](#)
- uint8\_t [persoUnblockRetries](#)
- uint8\_t [aidLength](#)
- uint8\_t [aidVal](#) [255]
- uint8\_t [univPin](#)
- uint8\_t [pin1State](#)
- uint8\_t [pin1Retries](#)
- uint8\_t [puk1Retries](#)
- uint8\_t [pin2State](#)
- uint8\_t [pin2Retries](#)
- uint8\_t [puk2Retries](#)

### 8.781.1 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[MAX_DESCRIPTION_LENGTH]</i>	<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>

## 8.781.2 Field Documentation

- 8.781.2.1    `uint8_t uim_appStatus::aidLength`
- 8.781.2.2    `uint8_t uim_appStatus::aidVal[255]`
- 8.781.2.3    `uint8_t uim_appStatus::appState`
- 8.781.2.4    `uint8_t uim_appStatus::appType`
- 8.781.2.5    `uint8_t uim_appStatus::persoFeature`
- 8.781.2.6    `uint8_t uim_appStatus::persoRetries`
- 8.781.2.7    `uint8_t uim_appStatus::persoState`
- 8.781.2.8    `uint8_t uim_appStatus::persoUnblockRetries`
- 8.781.2.9    `uint8_t uim_appStatus::pin1Retries`
- 8.781.2.10   `uint8_t uim_appStatus::pin1State`
- 8.781.2.11   `uint8_t uim_appStatus::pin2Retries`
- 8.781.2.12   `uint8_t uim_appStatus::pin2State`
- 8.781.2.13   `uint8_t uim_appStatus::puk1Retries`
- 8.781.2.14   `uint8_t uim_appStatus::puk2Retries`
- 8.781.2.15   `uint8_t uim_appStatus::univPin`

## 8.782   `uim_cardResult` Struct Reference

### Data Fields

- `uint8_t sw1`
- `uint8_t sw2`

### 8.782.1   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>

### 8.782.2   Field Documentation

- 8.782.2.1    `uint8_t uim_cardResult::sw1`

8.782.2.2 uint8\_t uim\_cardResult::sw2

## 8.783 uim\_cardStatus Struct Reference

### Data Fields

- uint16\_t [indexGwPri](#)
- uint16\_t [index1xPri](#)
- uint16\_t [indexGwSec](#)
- uint16\_t [index1xSec](#)
- uint8\_t [numSlot](#)
- [uim\\_slotInfo SlotInfo](#) [5]

### 8.783.1 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</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_slotInfo</a> for more information.</li> </ul>

## 8.783.2 Field Documentation

8.783.2.1 uint16\_t uim\_cardStatus::index1xPri

8.783.2.2 uint16\_t uim\_cardStatus::index1xSec

8.783.2.3 uint16\_t uim\_cardStatus::indexGwPri

8.783.2.4 uint16\_t uim\_cardStatus::indexGwSec

8.783.2.5 uint8\_t uim\_cardStatus::numSlot

8.783.2.6 uim\_slotInfo uim\_cardStatus::SlotInfo[5]

## 8.784 uim\_changeUIMPIN Struct Reference

### Data Fields

- uint8\_t [pinID](#)
- uint8\_t [oldPINLen](#)
- uint8\_t [oldPINVal](#) [255]
- uint8\_t [pinLen](#)
- uint8\_t [pinVal](#) [255]

### 8.784.1 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</i> [MAX_DESCRIPTION_LENGTH]	<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</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>

### 8.784.2 Field Documentation

8.784.2.1 uint8\_t uim\_changeUIMPIN::oldPINLen

8.784.2.2 uint8\_t uim\_changeUIMPIN::oldPINVal[255]

8.784.2.3 uint8\_t uim\_changeUIMPIN::pinID

8.784.2.4 uint8\_t uim\_changeUIMPIN::pinLen

8.784.2.5 uint8\_t uim\_changeUIMPIN::pinVal[255]

## 8.785 uim\_encryptedPIN1 Struct Reference

### Data Fields

- uint8\_t [pin1Len](#)
- uint8\_t [pin1Val](#) [255]

### 8.785.1 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.

### 8.785.2 Field Documentation

8.785.2.1 uint8\_t uim\_encryptedPIN1::pin1Len

8.785.2.2 uint8\_t uim\_encryptedPIN1::pin1Val[255]

## 8.786 uim\_fileInfo Struct Reference

### Data Fields

- uint16\_t [fileID](#)
- uint8\_t [pathLen](#)
- uint16\_t [path](#) [255]

### 8.786.1 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>

## 8.786.2 Field Documentation

8.786.2.1 uint16\_t uim\_fileInfo::fileID

8.786.2.2 uint16\_t uim\_fileInfo::path[255]

8.786.2.3 uint8\_t uim\_fileInfo::pathLen

## 8.787 uim\_hotSwapStatus Struct Reference

## Data Fields

- uint8\_t [hotSwapLength](#)
- uint8\_t [hotSwap](#) [255]

## 8.787.1 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</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>

## 8.787.2 Field Documentation

8.787.2.1 uint8\_t uim\_hotSwapStatus::hotSwap[255]

8.787.2.2 uint8\_t uim\_hotSwapStatus::hotSwapLength

## 8.788 uim\_readResult Struct Reference

## Data Fields

- uint16\_t [contentLen](#)
- uint8\_t [content](#) [255]

### 8.788.1 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> [255]	<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>

### 8.788.2 Field Documentation

8.788.2.1 uint8\_t uim\_readResult::content[255]

8.788.2.2 uint16\_t uim\_readResult::contentLen

## 8.789 uim\_readTransparentInfo Struct Reference

## Data Fields

- uint16\_t [offset](#)
- uint16\_t [length](#)

### 8.789.1 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>

### 8.789.2 Field Documentation

8.789.2.1 uint16\_t uim\_readTransparentInfo::length

8.789.2.2 uint16\_t uim\_readTransparentInfo::offset

## 8.790 uim\_remainingRetries Struct Reference

### Data Fields

- uint8\_t [verifyLeft](#)
- uint8\_t [unblockLeft](#)

### 8.790.1 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.

### 8.790.2 Field Documentation

8.790.2.1 uint8\_t uim\_remainingRetries::unblockLeft

8.790.2.2 uint8\_t uim\_remainingRetries::verifyLeft

## 8.791 uim\_sessionInformation Struct Reference

### Data Fields

- uint8\_t [sessionType](#)
- uint8\_t [aidLength](#)
- uint8\_t [aid](#) [255]

### 8.791.1 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>

## 8.791.2 Field Documentation

8.791.2.1 uint8\_t uim\_sessionInformation::aid[255]

8.791.2.2 uint8\_t uim\_sessionInformation::aidLength

8.791.2.3 uint8\_t uim\_sessionInformation::sessionType

## 8.792 uim\_setPINProtection Struct Reference

## Data Fields

- uint8\_t [pinID](#)
- uint8\_t [pinOperation](#)
- uint8\_t [pinLength](#)
- uint8\_t [pinValue](#) [255]

## 8.792.1 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>

## 8.792.2 Field Documentation

8.792.2.1 `uint8_t uim_setPINProtection::pinID`

8.792.2.2 `uint8_t uim_setPINProtection::pinLength`

8.792.2.3 `uint8_t uim_setPINProtection::pinOperation`

8.792.2.4 `uint8_t uim_setPINProtection::pinValue[255]`

## 8.793 uim\_slotInfo Struct Reference

### Data Fields

- `uint8_t cardState`
- `uint8_t upinState`
- `uint8_t upinRetries`
- `uint8_t upukRetries`
- `uint8_t errorState`
- `uint8_t numApp`
- `uim_appStatus AppStatus` [10]

### 8.793.1 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</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_appStatus</a> for more information.</li> </ul>

## 8.793.2 Field Documentation

8.793.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.793.2.2 `uint8_t uim_slotInfo::cardState`

8.793.2.3 `uint8_t uim_slotInfo::errorState`

8.793.2.4 `uint8_t uim_slotInfo::numApp`

8.793.2.5 `uint8_t uim_slotInfo::upinRetries`

8.793.2.6 `uint8_t uim_slotInfo::upinState`

8.793.2.7 `uint8_t uim_slotInfo::upukRetries`

## 8.794 `uim_UIMSessionInformation` Struct Reference

### Data Fields

- `uint8_t sessionType`
- `uint8_t aidLength`
- `uint8_t aid` [255]

### 8.794.1 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>

### 8.794.2 Field Documentation

8.794.2.1 `uint8_t uim_UIMSessionInformation::aid[255]`

8.794.2.2 `uint8_t uim_UIMSessionInformation::aidLength`

8.794.2.3 `uint8_t uim_UIMSessionInformation::sessionType`

## 8.795 `uim_unblockUIMPIN` Struct Reference

## Data Fields

- uint8\_t [pinID](#)
- uint8\_t [pukLen](#)
- uint8\_t [pukVal](#) [255]
- uint8\_t [newPINLen](#)
- uint8\_t [newPINVal](#) [255]

### 8.795.1 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[UIM_MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• PIN Unlock Key value.</li> <li>• This value is a sequence of ASCII characters.</li> </ul>
<i>newPINLen</i>	<ul style="list-style-type: none"> <li>• Length of the following elements i.e. new pin value.</li> </ul>
<i>newPINVal[UIM_MAX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• New PIN value.</li> <li>• This value is a sequence of ASCII characters.</li> </ul>

### 8.795.2 Field Documentation

8.795.2.1 uint8\_t uim\_unblockUIMPIN::newPINLen

8.795.2.2 uint8\_t uim\_unblockUIMPIN::newPINVal[255]

8.795.2.3 uint8\_t uim\_unblockUIMPIN::pinID

8.795.2.4 uint8\_t uim\_unblockUIMPIN::pukLen

8.795.2.5 uint8\_t uim\_unblockUIMPIN::pukVal[255]

## 8.796 uim\_verifyUIMPIN Struct Reference

## Data Fields

- uint8\_t [pinID](#)
- uint8\_t [pinLen](#)

- `uint8_t pinVal` [255]

### 8.796.1 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>

### 8.796.2 Field Documentation

8.796.2.1 `uint8_t uim_verifyUIMPIN::pinID`

8.796.2.2 `uint8_t uim_verifyUIMPIN::pinLen`

8.796.2.3 `uint8_t uim_verifyUIMPIN::pinVal`[255]

## 8.797 UIMAuthenticateReq Struct Reference

#### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [authenticationData authData](#)
- `ULONG * pIndicationToken`

### 8.797.1 Detailed Description

This structure contains information of the request parameters associated with a Authenticate 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="#">authData</a></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.

**8.797.2 Field Documentation**

**8.797.2.1 authenticationData** UIMAuthenticateReq::authData

**8.797.2.2 ULONG\*** UIMAuthenticateReq::pIndicationToken

**8.797.2.3 UIMSessionInformation** UIMAuthenticateReq::sessionInfo

**8.798 UIMAuthenticateResp Struct Reference****Data Fields**

- [cardResult](#) \* [pCardResult](#)
- [authenticateResult](#) \* [pAuthenticateResult](#)
- [ULONG](#) \* [pIndicationToken](#)

**8.798.1 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.

**8.798.2 Field Documentation**

**8.798.2.1 authenticateResult\*** UIMAuthenticateResp::pAuthenticateResult

**8.798.2.2 cardResult\*** UIMAuthenticateResp::pCardResult

8.798.2.3 **ULONG\*** `UIMAuthenticateResp::pIndicationToken`

## 8.799 UIMChangePinReq Struct Reference

### Data Fields

- [UIMSessionInformation](#) `sessionInfo`
- [changeUIMPIN](#) `changePIN`
- **BYTE \*** `pKeyReferenceID`
- **ULONG \*** `pIndicationToken`

### 8.799.1 Detailed Description

This structure contains information of the request parameters associated with a Change 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="#">changePIN</a>	<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.

### 8.799.2 Field Documentation

8.799.2.1 **changeUIMPIN** `UIMChangePinReq::changePIN`

8.799.2.2 **ULONG\*** `UIMChangePinReq::pIndicationToken`

8.799.2.3 **BYTE\*** `UIMChangePinReq::pKeyReferenceID`

8.799.2.4 **UIMSessionInformation** `UIMChangePinReq::sessionInfo`

## 8.800 UIMDepersonalizationReq Struct Reference

### Data Fields

- [depersonalizationInformation](#) `depersonalisationInfo`

### 8.800.1 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>
-------------------------------	--

### 8.800.2 Field Documentation

8.800.2.1 `depersonalizationInformation` `UIMDepersonalizationReq::depersonalisationInfo`

## 8.801 UIMDepersonalizationResp Struct Reference

#### Data Fields

- [remainingRetries](#) \* [pRemainingRetries](#)

### 8.801.1 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.

### 8.801.2 Field Documentation

8.801.2.1 `remainingRetries*` `UIMDepersonalizationResp::pRemainingRetries`

## 8.802 UIMEventRegisterReqResp Struct Reference

#### Data Fields

- [ULONG](#) `eventMask`

### 8.802.1 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> <li>– Bit 4 - Physical Slot Status</li> </ul> </li> </ul>
------------------------------	--

## 8.802.2 Field Documentation

## 8.802.2.1 ULONG UIMEventRegisterReqResp::eventMask

## 8.803 UIMGetCardStatusResp Struct Reference

## Data Fields

- [cardStatus](#) \* [pCardStatus](#)
- [hotSwapStatus](#) \* [pHotSwapStatus](#)

## 8.803.1 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.

## 8.803.2 Field Documentation

## 8.803.2.1 cardStatus\* UIMGetCardStatusResp::pCardStatus

## 8.803.2.2 hotSwapStatus\* UIMGetCardStatusResp::pHotSwapStatus

## 8.804 UIMGetConfigurationReq Struct Reference

## Data Fields

- [ULONG](#) \* [pConfigurationMask](#)

### 8.804.1 Detailed Description

This structure contains information of the request parameters associated with to gets the modem configuration for the UIM module API.

#### Parameters

<i>pConfigurationMask(optional)</i>	<ul style="list-style-type: none"> <li>Requested configurations             <ul style="list-style-type: none"> <li>– Bit 0 - Automatic selection</li> <li>– Bit 1 - Personalization status</li> <li>– Bit 2 - Halt subscription</li> <li>– All other bits are reserved for future use</li> </ul> </li> </ul>
-------------------------------------	--

#### Note

- if the TLV is missing, the service returns all configuration items in the response.

### 8.804.2 Field Documentation

#### 8.804.2.1 ULONG\* UIMGetConfigurationReq::pConfigurationMask

## 8.805 UIMGetConfigurationResp Struct Reference

#### Data Fields

- [BYTE](#) \* [pAutoSelection](#)
- [personalizationStatus](#) \* [pPersonalizationStatus](#)
- [BYTE](#) \* [pHaltSubscription](#)

### 8.805.1 Detailed Description

This structure contains information of the response parameters associated with a Read Transparent API.

#### Parameters

<i>pAutoSelection(optional)</i>	<ul style="list-style-type: none"> <li>Indicates whether the modem is configured to automatically select the provisioning sessions at powerup.</li> <li>Valid values             <ul style="list-style-type: none"> <li>– 0 - Automatic provisioning is off</li> <li>– 1 - Automatic provisioning is on</li> </ul> </li> </ul>
<i>pPersonalizationStatus(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">personalizationStatus</a> for more information.</li> </ul>
<i>pHaltSubscription(optional)</i>	<ul style="list-style-type: none"> <li>Indicates if the modem is configured to publish the subscription after successful initialization.</li> <li>Valid values             <ul style="list-style-type: none"> <li>– 0 - Modem proceeds with publishing the subscription</li> <li>– 1 - Modem does not publish the subscription</li> </ul> </li> </ul>

## 8.805.2 Field Documentation

8.805.2.1 **BYTE\*** `UIMGetConfigurationResp::pAutoSelection`

8.805.2.2 **BYTE\*** `UIMGetConfigurationResp::pHaltSubscription`

8.805.2.3 **personalizationStatus\*** `UIMGetConfigurationResp::pPersonalizationStatus`

## 8.806 UIMGetFileAttributesReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [fileInfo fileIndex](#)
- **ULONG \*** `pIndicationToken`

### 8.806.1 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.

## 8.806.2 Field Documentation

8.806.2.1 **fileInfo** `UIMGetFileAttributesReq::fileIndex`

8.806.2.2 **ULONG\*** `UIMGetFileAttributesReq::pIndicationToken`

8.806.2.3 **UIMSessionInformation** `UIMGetFileAttributesReq::sessionInfo`

## 8.807 UIMGetFileAttributesResp Struct Reference

### Data Fields

- **cardResult \*** `pCardResult`
- **fileAttributes \*** `pFileAttributes`
- **ULONG \*** `pIndicationToken`

### 8.807.1 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.

### 8.807.2 Field Documentation

8.807.2.1 **cardResult\*** UIMGetFileAttributesResp::pCardResult

8.807.2.2 **fileAttributes\*** UIMGetFileAttributesResp::pFileAttributes

8.807.2.3 **ULONG\*** UIMGetFileAttributesResp::pIndicationToken

## 8.808 UIMGetSlotsStatusResp Struct Reference

#### Data Fields

- BYTE \*** [pNumberOfPhySlot](#)
- UIMSlotsStatus \*** [pUimSlotsStatus](#)

### 8.808.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>pNumberOfPhy-Slot</i>	<ul style="list-style-type: none"><li>Number of sets of the Slot Status.</li></ul>
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"><li>Slots Status See <a href="#">UIMSlotsStatus</a> for more information..</li></ul>

### 8.808.2 Field Documentation

8.808.2.1 **BYTE\*** UIMGetSlotsStatusResp::pNumberOfPhySlot

8.808.2.2 UIMSlotsStatus\* UIMGetSlotsStatusResp::pUimSlotsStatus

## 8.809 UIMPinResp Struct Reference

### Data Fields

- [remainingRetries](#) \* [pRemainingRetries](#)
- [encryptedPIN1](#) \* [pEncryptedPIN1](#)
- [ULONG](#) \* [pIndicationToken](#)

### 8.809.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

#### Parameters

<i>pRemainingRetries(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>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 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.

### 8.809.2 Field Documentation

8.809.2.1 encryptedPIN1\* UIMPinResp::pEncryptedPIN1

8.809.2.2 ULONG\* UIMPinResp::pIndicationToken

8.809.2.3 remainingRetries\* UIMPinResp::pRemainingRetries

## 8.810 UIMPowerDownReq Struct Reference

### Data Fields

- [BYTE](#) slot

### 8.810.1 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>
-------------	--

## 8.810.2 Field Documentation

## 8.810.2.1 BYTE UIMPowerDownReq::slot

## 8.811 UIMPowerUpReq Struct Reference

## Data Fields

- BYTE slot
- BYTE \* plgnoreHotSwapSwitch

## 8.811.1 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>

## 8.811.2 Field Documentation

## 8.811.2.1 BYTE\* UIMPowerUpReq::plgnoreHotSwapSwitch

## 8.811.2.2 BYTE UIMPowerUpReq::slot

## 8.812 UIMReadTransparentReq Struct Reference

## Data Fields

- UIMSessionInformation sessionInfo
- fileInfo fileIndex
- readTransparentInfo readTransparent
- ULONG \* pIndicationToken
- BYTE \* pEncryptData

### 8.812.1 Detailed Description

This structure contains information of the request parameters associated with a Read Transparent API.

#### Parameters

<i>sessionInfo</i>	<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.

### 8.812.2 Field Documentation

8.812.2.1 **fileInfo** `UIMReadTransparentReq::fileIndex`

8.812.2.2 **BYTE\*** `UIMReadTransparentReq::pEncryptData`

8.812.2.3 **ULONG\*** `UIMReadTransparentReq::pIndicationToken`

8.812.2.4 **readTransparentInfo** `UIMReadTransparentReq::readTransparent`

8.812.2.5 **UIMSessionInformation** `UIMReadTransparentReq::sessionInfo`

## 8.813 UIMReadTransparentResp Struct Reference

#### Data Fields

- [cardResult](#) \* [pCardResult](#)
- [readResult](#) \* [pReadResult](#)
- [ULONG](#) \* [pIndicationToken](#)
- [BYTE](#) \* [pEncryptedData](#)

### 8.813.1 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.

## 8.813.2 Field Documentation

8.813.2.1 **cardResult\*** UIMReadTransparentResp::pCardResult

8.813.2.2 **BYTE\*** UIMReadTransparentResp::pEncryptedData

8.813.2.3 **ULONG\*** UIMReadTransparentResp::pIndicationToken

8.813.2.4 **readResult\*** UIMReadTransparentResp::pReadResult

## 8.814 UIMRefreshCompleteReq Struct Reference

## Data Fields

- [UIMSessionInformation sessionInfo](#)
- [BYTE refreshComplete](#)

## 8.814.1 Detailed Description

This structure contains information of the request parameters associated with a SLQSUIMRefreshComplete.

## Parameters

<i><a href="#">sessionInfo(-Mandatory)</a></i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
<i><a href="#">refresh-Complete(-Mandatory)</a></i>	<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>

### 8.814.2 Field Documentation

8.814.2.1 **BYTE** UIMRefreshCompleteReq::refreshComplete

8.814.2.2 **UIMSessionInformation** UIMRefreshCompleteReq::sessionInfo

## 8.815 UIMRefreshEvent Struct Reference

### Data Fields

- [BYTE](#) stage
- [BYTE](#) mode
- [BYTE](#) sessionType
- [BYTE](#) aidLength
- [BYTE](#) aid [255]
- [WORD](#) numOfFiles
- [fileInfo](#) arrfileInfo [255]

### 8.815.1 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>

## 8.815.2 Field Documentation

8.815.2.1 **BYTE** UIMRefreshEvent::aid[255]

8.815.2.2 **BYTE** UIMRefreshEvent::aidLength

8.815.2.3 **fileInfo** UIMRefreshEvent::arrfileInfo[255]

8.815.2.4 **BYTE** UIMRefreshEvent::mode

8.815.2.5 **WORD** UIMRefreshEvent::numOfFiles

8.815.2.6 **BYTE** UIMRefreshEvent::sessionType

8.815.2.7 **BYTE** UIMRefreshEvent::stage

## 8.816 UIMRefreshGetLastEventReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)

### 8.816.1 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>
--	---

### 8.816.2 Field Documentation

8.816.2.1 [UIMSessionInformation](#) [UIMRefreshGetLastEventReq::sessionInfo](#)

## 8.817 UIMRefreshGetLastEventResp Struct Reference

#### Data Fields

- [UIMRefreshEvent](#) \* [pRefreshEvent](#)

### 8.817.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIMRefreshGetLastEvent.

#### Parameters

<i><a href="#">refreshEvent</a>(-Optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshEvent</a> for more information.</li> </ul>
--	---

### 8.817.2 Field Documentation

8.817.2.1 [UIMRefreshEvent](#)\* [UIMRefreshGetLastEventResp::pRefreshEvent](#)

## 8.818 UIMRefreshOKReq Struct Reference

#### Data Fields

- [UIMSessionInformation](#) [sessionInfo](#)
- [BYTE](#) [OKtoRefresh](#)

### 8.818.1 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>OKtoRefresh</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>
--------------------	---

## 8.818.2 Field Documentation

8.818.2.1 **BYTE** UIMRefreshOKReq::OKtoRefresh

8.818.2.2 **UIMSessionInformation** UIMRefreshOKReq::sessionInfo

## 8.819 UIMRefreshRegisterReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [registerRefresh regRefresh](#)

### 8.819.1 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>

## 8.819.2 Field Documentation

8.819.2.1 **registerRefresh** UIMRefreshRegisterReq::regRefresh

8.819.2.2 **UIMSessionInformation** UIMRefreshRegisterReq::sessionInfo

## 8.820 UIMSessionInformation Struct Reference

### Data Fields

- BYTE** [sessionType](#)
- BYTE** [aidLength](#)
- BYTE** [aid](#) [255]

### 8.820.1 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>

### 8.820.2 Field Documentation

8.820.2.1 **BYTE** `UIMSessionInformation::aid[255]`

8.820.2.2 **BYTE** `UIMSessionInformation::aidLength`

8.820.2.3 **BYTE** `UIMSessionInformation::sessionType`

## 8.821 UIMSetPinProtectionReq Struct Reference

### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [setPINProtection pinProtection](#)
- BYTE** \* [pKeyReferenceID](#)
- ULONG** \* [pIndicationToken](#)

### 8.821.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

#### Parameters

<a href="#">sessionInfo</a>	<ul style="list-style-type: none"> <li>See <a href="#">UIMSessionInformation</a> for more information.</li> </ul>
-----------------------------	---

<i>pinProtection</i>	<ul style="list-style-type: none"> <li>See <a href="#">setPINProtection</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.

**8.821.2 Field Documentation**

8.821.2.1 **ULONG\*** `UIMSetPinProtectionReq::pIndicationToken`

8.821.2.2 **setPINProtection** `UIMSetPinProtectionReq::pinProtection`

8.821.2.3 **BYTE\*** `UIMSetPinProtectionReq::pKeyReferenceID`

8.821.2.4 **UIMSessionInformation** `UIMSetPinProtectionReq::sessionInfo`

**8.822 UIMSlotsStatus Struct Reference****Data Fields**

- [UIMSlotStatus](#) `uimSlotStatus` [255]

**8.822.1 Detailed Description**

This structure contains information of the response parameters associated with a Get Slots Status API.

**Parameters**

<i>uimSlotStatus[ MAX_SLOTS_S- TATUS]</i>	<ul style="list-style-type: none"> <li>Contain all slots status.</li> </ul>
---	---

**8.822.2 Field Documentation**

8.822.2.1 **UIMSlotStatus** `UIMSlotsStatus::uimSlotStatus[255]`

**8.823 UIMSlotStatus Struct Reference**

## Data Fields

- [ULONG uPhyCardStatus](#)
- [ULONG uPhySlotStatus](#)
- [BYTE bLogicalSlot](#)
- [BYTE bICCIDLength](#)
- [BYTE bICCID \[255\]](#)

### 8.823.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>uPhyCardStatus</i>	<ul style="list-style-type: none"> <li>• State of the card in the Pyhsical Slot Status. <ul style="list-style-type: none"> <li>– 0x00 - Unknown.</li> <li>– 0x01 - Absent.</li> <li>– 0x02 - Present.</li> </ul> </li> </ul>
<i>uPhySlotStatus</i>	<ul style="list-style-type: none"> <li>• State of the Physical Slot status. <ul style="list-style-type: none"> <li>– 0x00 Inactive.</li> <li>– 0x01 Activate.</li> </ul> </li> </ul>
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> <li>• Logical Slot associated with this physical slot. THis is valid if the physical slot is active. <ul style="list-style-type: none"> <li>– 1 - Slot 1.</li> <li>– 2 - Slot 2.</li> <li>– 3 - Slot 3.</li> <li>– 4 - Slot 4.</li> <li>– 5 - Slot 5.</li> </ul> </li> </ul>
<i>bLogicalSlot</i>	<ul style="list-style-type: none"> <li>• Number of sets the sets of ICCID</li> </ul>
<i>bICCID[MAX_ICCID_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Contains the ICCID of the card in the physical slot.</li> </ul>

### 8.823.2 Field Documentation

8.823.2.1 **BYTE** UIMSlotStatus::bICCID[255]

8.823.2.2 **BYTE** UIMSlotStatus::bICCIDLength

8.823.2.3 **BYTE** UIMSlotStatus::bLogicalSlot

8.823.2.4 **ULONG** UIMSlotStatus::uPhyCardStatus

8.823.2.5 **ULONG** UIMSlotStatus::uPhySlotStatus

## 8.824 UIMSlotStatusChangeInfo Struct Reference

### Data Fields

- [UIMSlotsStatus slotsstatusChange](#)
- [BYTE bNumberOfPhySlots](#)

### 8.824.1 Detailed Description

Structure consist of cardstatus params

#### Parameters

<i>slotstatus- Change</i>	<ul style="list-style-type: none"><li>• See <a href="#">UIMSlotStatus</a> for more information</li></ul>
<i>bNumberOfPhy- Slots</i>	<ul style="list-style-type: none"><li>• Number of Physical Slot(s)</li></ul>

### 8.824.2 Field Documentation

8.824.2.1 [BYTE UIMSlotStatusChangeInfo::bNumberOfPhySlots](#)

8.824.2.2 [UIMSlotsStatus UIMSlotStatusChangeInfo::slotsstatusChange](#)

## 8.825 UIMStatusChangeInfo Struct Reference

### Data Fields

- [cardStatus statusChange](#)

### 8.825.1 Detailed Description

Structure consist of cardstatus params

#### Parameters

<i>statusChange</i>	<ul style="list-style-type: none"><li>• See <a href="#">cardStatus</a> for more information</li></ul>
---------------------	---

### 8.825.2 Field Documentation

8.825.2.1 [cardStatus UIMStatusChangeInfo::statusChange](#)

## 8.826 UIMSwitchSlotReq Struct Reference

### Data Fields

- [BYTE bLogicalSlot](#)
- [ULONG ulPhysicalSlot](#)

### 8.826.1 Detailed Description

This structure contains information of the request parameters associated with a Switch Slot.

#### Parameters

<i>bLogicalSlot</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> <li>3 - Slot 3</li> <li>4 - Slot 4</li> <li>5 - Slot 5</li> </ul> </li> </ul>
<i>bPhysicalSlot</i>	<ul style="list-style-type: none"> <li>1 - Slot 1</li> <li>2 - Slot 2</li> <li>3 - Slot 3</li> <li>4 - Slot 4</li> <li>5 - Slot 5</li> </ul>

### 8.826.2 Field Documentation

8.826.2.1 **BYTE** UIMSwitchSlotReq::bLogicalSlot

8.826.2.2 **ULONG** UIMSwitchSlotReq::ulPhysicalSlot

## 8.827 UIMUnblockPinReq Struct Reference

#### Data Fields

- [UIMSessionInformation sessionInfo](#)
- [unblockUIMPIN unblockPIN](#)
- BYTE** \* pKeyReferenceID
- ULONG** \* pIndicationToken

### 8.827.1 Detailed Description

This structure contains information of the request parameters associated with a Unblock PIN 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="#">unblockPIN</a></i>	<ul style="list-style-type: none"> <li>See <a href="#">unblockUIMPIN</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.

**8.827.2 Field Documentation**

**8.827.2.1** **ULONG\*** UIMUnblockPinReq::pIndicationToken

**8.827.2.2** **BYTE\*** UIMUnblockPinReq::pKeyReferenceID

**8.827.2.3** **UIMSessionInformation** UIMUnblockPinReq::sessionInfo

**8.827.2.4** **unblockUIMPIN** UIMUnblockPinReq::unblockPIN

**8.828 UIMVerifyPinReq Struct Reference****Data Fields**

- [UIMSessionInformation sessionInfo](#)
- [verifyUIMPIN verifyPIN](#)
- [encryptedPIN1 \\* pEncryptedPIN1](#)
- [BYTE \\* pKeyReferenceID](#)
- [ULONG \\* pIndicationToken](#)

**8.828.1 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>

<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.

**8.828.2 Field Documentation**

**8.828.2.1 encryptedPIN1** \* UIMVerifyPinReq::pEncryptedPIN1

**8.828.2.2 ULONG** \* UIMVerifyPinReq::pIndicationToken

**8.828.2.3 BYTE** \* UIMVerifyPinReq::pKeyReferenceID

**8.828.2.4 UIMSessionInformation** UIMVerifyPinReq::sessionInfo

**8.828.2.5 verifyUIMPIN** UIMVerifyPinReq::verifyPIN

**8.829 UMTSInfo Struct Reference****Data Fields**

- [WORD cellID](#)
- [BYTE plmn](#) [3]
- [WORD lac](#)
- [WORD uarfcn](#)
- [WORD psc](#)
- [SHORT rscn](#)
- [SHORT ecio](#)
- [BYTE umtsInst](#)
- [UMTSInstInfo UMTSInstInfo](#) [255]
- [BYTE geranInst](#)
- [geranInstInfo GeranInstInfo](#) [255]

**8.829.1 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> [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>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</i> [MAX_DESCRIPTOR_LENGTH]	<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</i> [MAX_DESCRIPTOR_LENGTH]	<ul style="list-style-type: none"> <li>• See <a href="#">geranInstInfo</a> for more information.</li> </ul>

## 8.829.2 Field Documentation

### 8.829.2.1 WORD UMTSInfo::cellID

### 8.829.2.2 SHORT UMTSInfo::ecio

### 8.829.2.3 BYTE UMTSInfo::geranInst

### 8.829.2.4 [geranInstInfo](#) UMTSInfo::GeranInstInfo[255]

### 8.829.2.5 WORD UMTSInfo::lac

8.829.2.6 **BYTE** UMTSInfo::plmn[3]

8.829.2.7 **WORD** UMTSInfo::psc

8.829.2.8 **SHORT** UMTSInfo::rscp

8.829.2.9 **WORD** UMTSInfo::uarfcn

8.829.2.10 **BYTE** UMTSInfo::umtsInst

8.829.2.11 **UMTSInstInfo** UMTSInfo::UMTSInstInfo[255]

## 8.830 UMTSInstInfo Struct Reference

### Data Fields

- [WORD umtsUarfcn](#)
- [WORD umtsPsc](#)
- [SHORT umtsRscp](#)
- [SHORT umtsEcio](#)

### 8.830.1 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>

### 8.830.2 Field Documentation

8.830.2.1 **SHORT** UMTSInstInfo::umtsEcio

8.830.2.2 **WORD** UMTSInstInfo::umtsPsc

8.830.2.3 **SHORT** UMTSInstInfo::umtsRscp

8.830.2.4 **WORD** UMTSInstInfo::umtsUarfcn

## 8.831 umtsLTENbrCell Struct Reference

## Data Fields

- [WORD earfcn](#)
- [WORD pci](#)
- [ULONG rsrp](#)
- [ULONG rsrq](#)
- [SHORT srxlev](#)
- [BYTE cellsTDD](#)

### 8.831.1 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_CEL_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>

### 8.831.2 Field Documentation

8.831.2.1 **BYTE** umtsLTENbrCell::cellsTDD

8.831.2.2 **WORD** umtsLTENbrCell::earfcn

8.831.2.3 **WORD** umtsLTENbrCell::pci

8.831.2.4 **ULONG** umtsLTENbrCell::rsrp

8.831.2.5 **ULONG** umtsLTENbrCell::rsrq

8.831.2.6 **SHORT** umtsLTENbrCell::srxlev

## 8.832 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

### 8.832.1 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 \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 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>	<ul style="list-style-type: none"> <li>- delivery of erroneous SDUs</li> <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>	<ul style="list-style-type: none"> <li>- Transfer delay (ms)</li> <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>	<ul style="list-style-type: none"> <li>- Transfer handling priority</li> <li>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

## 8.832.2 Field Documentation

### 8.832.2.1 BYTE UMTSMinQoS::deliveryErrSDU

8.832.2.2 **ULONG** UMTSMinQoS::grntDownlinkBitrate

8.832.2.3 **ULONG** UMTSMinQoS::grntUplinkBitrate

8.832.2.4 **ULONG** UMTSMinQoS::maxDownlinkBitrate

8.832.2.5 **ULONG** UMTSMinQoS::maxSDUSize

8.832.2.6 **ULONG** UMTSMinQoS::maxUplinkBitrate

8.832.2.7 **BYTE** UMTSMinQoS::qosDeliveryOrder

8.832.2.8 **BYTE** UMTSMinQoS::resBerRatio

8.832.2.9 **BYTE** UMTSMinQoS::sduErrorRatio

8.832.2.10 **BYTE** UMTSMinQoS::trafficClass

8.832.2.11 **ULONG** UMTSMinQoS::trafficPriority

8.832.2.12 **ULONG** UMTSMinQoS::transferDelay

## 8.833 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](#)

### 8.833.1 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 \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>	<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 \times 10^{-2}</math></li> <li>• 0x02 - <math>1 \times 10^{-2}</math></li> <li>• 0x03 - <math>5 \times 10^{-3}</math></li> <li>• 0x04 - <math>4 \times 10^{-3}</math></li> <li>• 0x05 - <math>1 \times 10^{-3}</math></li> <li>• 0x06 - <math>1 \times 10^{-4}</math></li> <li>• 0x07 - <math>1 \times 10^{-5}</math></li> <li>• 0x08 - <math>1 \times 10^{-6}</math></li> <li>• 0x09 - <math>1 \times 10^{-8}</math></li> </ul>

<i>deliveryErrSDU</i>	<ul style="list-style-type: none"> <li>- Delivery of erroneous SDUs</li> <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>	<ul style="list-style-type: none"> <li>- Transfer delay (ms)</li> <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>	<ul style="list-style-type: none"> <li>- Transfer handling priority</li> <li>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

### 8.833.2 Field Documentation

8.833.2.1 BYTE UMTSQoS::deliveryErrSDU

8.833.2.2 ULONG UMTSQoS::grntDownlinkBitrate

8.833.2.3 ULONG UMTSQoS::grntUplinkBitrate

8.833.2.4 ULONG UMTSQoS::maxDownlinkBitrate

8.833.2.5 ULONG UMTSQoS::maxSDUSize

8.833.2.6 ULONG UMTSQoS::maxUplinkBitrate

8.833.2.7 BYTE UMTSQoS::qosDeliveryOrder

8.833.2.8 BYTE UMTSQoS::resBerRatio

8.833.2.9 BYTE UMTSQoS::sduErrorRatio

8.833.2.10 BYTE UMTSQoS::trafficClass

8.833.2.11 ULONG UMTSQoS::trafficPriority

8.833.2.12 ULONG UMTSQoS::transferDelay

### 8.834 UMTSReqQoSsigInd Struct Reference

## Data Fields

- struct [UMTSQoS UMTSReqQoS](#)
- [BYTE SigInd](#)

### 8.834.1 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>

### 8.834.2 Field Documentation

8.834.2.1 [BYTE UMTSReqQoSSigInd::SigInd](#)

8.834.2.2 [struct UMTSQoS UMTSReqQoSSigInd::UMTSReqQoS](#)

## 8.835 unblockUIMPIN Struct Reference

## Data Fields

- [BYTE pinID](#)
- [BYTE pukLen](#)
- [BYTE pukVal](#) [255]
- [BYTE newPINLen](#)
- [BYTE newPINVal](#) [255]

### 8.835.1 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>

## 8.835.2 Field Documentation

8.835.2.1 **BYTE** unblockUIMPIN::newPINLen

8.835.2.2 **BYTE** unblockUIMPIN::newPINVal[255]

8.835.2.3 **BYTE** unblockUIMPIN::pinID

8.835.2.4 **BYTE** unblockUIMPIN::pukLen

8.835.2.5 **BYTE** unblockUIMPIN::pukVal[255]

## 8.836 UniversalTime Struct Reference

### Data Fields

- [WORD](#) year
- [BYTE](#) month
- [BYTE](#) day
- [BYTE](#) hour
- [BYTE](#) minute
- [BYTE](#) second
- [BYTE](#) dayOfWeek

### 8.836.1 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>

## 8.836.2 Field Documentation

8.836.2.1 BYTE UniversalTime::day

8.836.2.2 BYTE UniversalTime::dayOfWeek

8.836.2.3 BYTE UniversalTime::hour

8.836.2.4 BYTE UniversalTime::minute

8.836.2.5 BYTE UniversalTime::month

8.836.2.6 BYTE UniversalTime::second

8.836.2.7 WORD UniversalTime::year

## 8.837 unpack\_dms\_GetActivationState\_t Struct Reference

### Data Fields

- uint8\_t [state](#)

### 8.837.1 Detailed Description

## Parameters

<i>pActivation-State[OUT]</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>
-------------------------------	--

**8.837.2 Field Documentation**

8.837.2.1 uint8\_t unpack\_dms\_GetActivationState\_t::state

**8.838 unpack\_dms\_GetBandCapability\_t Struct Reference****Data Fields**

- uint32\_t [BandCapability](#)
- uint16\_t [Tlvresult](#)

**8.838.1 Field Documentation**

8.838.1.1 uint32\_t unpack\_dms\_GetBandCapability\_t::BandCapability

8.838.1.2 uint16\_t unpack\_dms\_GetBandCapability\_t::Tlvresult

**8.839 unpack\_dms\_GetCrashAction\_t Struct Reference****Data Fields**

- uint8\_t [DevCrashState](#)
- uint16\_t [Tlvresult](#)

**8.839.1 Field Documentation**

8.839.1.1 uint8\_t unpack\_dms\_GetCrashAction\_t::DevCrashState

8.839.1.2 uint16\_t unpack\_dms\_GetCrashAction\_t::Tlvresult

**8.840 unpack\_dms\_GetCustFeature\_t Struct Reference****Data Fields**

- uint32\_t [GpsEnable](#)
- uint8\_t [DisableIMSI](#)

- uint16\_t [IPFamSupport](#)
- uint8\_t [RMAutoConnect](#)
- uint8\_t [GPSSel](#)
- uint8\_t [SMSSupport](#)
- uint8\_t [IsVoiceEnabled](#)
- uint8\_t [DHCPRelayEnabled](#)
- uint8\_t [GPSLPM](#)
- uint16\_t [Tlvresult](#)

### 8.840.1 Field Documentation

8.840.1.1 uint8\_t unpack\_dms\_GetCustFeature\_t::DHCPRelayEnabled

8.840.1.2 uint8\_t unpack\_dms\_GetCustFeature\_t::DisableIMSI

8.840.1.3 uint32\_t unpack\_dms\_GetCustFeature\_t::GpsEnable

8.840.1.4 uint8\_t unpack\_dms\_GetCustFeature\_t::GPSLPM

8.840.1.5 uint8\_t unpack\_dms\_GetCustFeature\_t::GPSSel

8.840.1.6 uint16\_t unpack\_dms\_GetCustFeature\_t::IPFamSupport

8.840.1.7 uint8\_t unpack\_dms\_GetCustFeature\_t::IsVoiceEnabled

8.840.1.8 uint8\_t unpack\_dms\_GetCustFeature\_t::RMAutoConnect

8.840.1.9 uint8\_t unpack\_dms\_GetCustFeature\_t::SMSSupport

8.840.1.10 uint16\_t unpack\_dms\_GetCustFeature\_t::Tlvresult

## 8.841 unpack\_dms\_GetCustFeaturesV2\_t Struct Reference

### Data Fields

- [DMSgetCustomFeatureV2](#) [GetCustomFeatureV2](#)
- uint16\_t [Tlvresult](#)

### 8.841.1 Detailed Description

This structure contains customization settings set to modem unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
------------------	---

### 8.841.2 Field Documentation

8.841.2.1 [DMSgetCustomFeatureV2](#) [unpack\\_dms\\_GetCustFeaturesV2\\_t::GetCustomFeatureV2](#)

8.841.2.2 uint16\_t [unpack\\_dms\\_GetCustFeaturesV2\\_t::Tlvresult](#)

## 8.842 unpack\_dms\_GetDeviceCap\_t Struct Reference

### Data Fields

- uint32\_t [MaxTXChannelRate](#)
- uint32\_t [MaxRXChannelRate](#)
- uint32\_t [DataServiceCapability](#)
- uint32\_t [SimCapability](#)
- uint32\_t [RadiolfacesSize](#)
- uint8\_t [Radiolfaces](#) [64]
- uint16\_t [Tlvresult](#)

### 8.842.1 Field Documentation

8.842.1.1 uint32\_t unpack\_dms\_GetDeviceCap\_t::DataServiceCapability

8.842.1.2 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxRXChannelRate

8.842.1.3 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxTXChannelRate

8.842.1.4 uint8\_t unpack\_dms\_GetDeviceCap\_t::Radiolfaces[64]

8.842.1.5 uint32\_t unpack\_dms\_GetDeviceCap\_t::RadiolfacesSize

8.842.1.6 uint32\_t unpack\_dms\_GetDeviceCap\_t::SimCapability

8.842.1.7 uint16\_t unpack\_dms\_GetDeviceCap\_t::Tlvresult

## 8.843 unpack\_dms\_GetDeviceCapabilities\_t Struct Reference

### Data Fields

- uint32\_t [maxTxChannelRate](#)
- uint32\_t [maxRxChannelRate](#)
- uint32\_t [dataServiceCaCapability](#)
- uint32\_t [simCapability](#)
- uint32\_t [radiolfacesSize](#)
- uint8\_t [Radiolfaces](#) [255]

### 8.843.1 Detailed Description

#### Parameters

<i>maxTxChannelRate</i>	Maximum Tx transmission rate in bits per second.
<i>maxRxChannelRate</i>	Maximum Rx transmission rate in bits per second
<i>dataServiceCaCapability</i>	data service capability
<i>simCapability</i>	SIM Capability
<i>radiolfacesSize</i>	radio interface length
<i>Radiolfaces</i>	radio interfaces

### 8.843.2 Field Documentation

8.843.2.1 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::dataServiceCaCapability

8.843.2.2 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::maxRxChannelRate

8.843.2.3 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::maxTxChannelRate

8.843.2.4 uint8\_t unpack\_dms\_GetDeviceCapabilities\_t::Radiofaces[255]

8.843.2.5 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::radiofacesSize

8.843.2.6 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::simCapability

## 8.844 unpack\_dms\_GetDeviceHardwareRev\_t Struct Reference

### Data Fields

- uint8\_t [stringSize](#)
- char [String](#) [255]
- uint16\_t [Tlvresult](#)

### 8.844.1 Field Documentation

8.844.1.1 char unpack\_dms\_GetDeviceHardwareRev\_t::String[255]

8.844.1.2 uint8\_t unpack\_dms\_GetDeviceHardwareRev\_t::stringSize

8.844.1.3 uint16\_t unpack\_dms\_GetDeviceHardwareRev\_t::Tlvresult

## 8.845 unpack\_dms\_GetDeviceMfr\_t Struct Reference

### Data Fields

- uint8\_t [stringSize](#)
- char [String](#) [255]
- uint16\_t [Tlvresult](#)

### 8.845.1 Field Documentation

8.845.1.1 char unpack\_dms\_GetDeviceMfr\_t::String[255]

8.845.1.2 uint8\_t unpack\_dms\_GetDeviceMfr\_t::stringSize

8.845.1.3 uint16\_t unpack\_dms\_GetDeviceMfr\_t::Tlvresult

## 8.846 unpack\_dms\_GetDeviceSerialNumbers\_t Struct Reference

### Data Fields

- uint8\_t [esnSize](#)
- char [ESNString](#) [255]

- uint8\_t [imeiSize](#)
- char [IMEIString](#) [255]
- uint8\_t [meidSize](#)
- char [MEIDString](#) [255]
- uint8\_t [imeiSvnSize](#)
- char [ImeiSvnString](#) [255]
- uint16\_t [Tlvresult](#)

### 8.846.1 Field Documentation

8.846.1.1 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::esnSize

8.846.1.2 char unpack\_dms\_GetDeviceSerialNumbers\_t::ESNString[255]

8.846.1.3 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::imeiSize

8.846.1.4 char unpack\_dms\_GetDeviceSerialNumbers\_t::IMEIString[255]

8.846.1.5 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::imeiSvnSize

8.846.1.6 char unpack\_dms\_GetDeviceSerialNumbers\_t::ImeiSvnString[255]

8.846.1.7 uint8\_t unpack\_dms\_GetDeviceSerialNumbers\_t::meidSize

8.846.1.8 char unpack\_dms\_GetDeviceSerialNumbers\_t::MEIDString[255]

8.846.1.9 uint16\_t unpack\_dms\_GetDeviceSerialNumbers\_t::Tlvresult

## 8.847 unpack\_dms\_GetFirmwareInfo\_t 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]
- uint16\_t [Tlvresult](#)

### 8.847.1 Detailed Description

#### Parameters

<i>modelid_str</i>	Mode ID String.
<i>bootversion_str</i>	Boot Version.
<i>appversion_str</i>	Application Version String.
<i>sku_str</i>	SKU String.
<i>packageid_str</i>	<ul style="list-style-type: none"> <li>• Package ID String.</li> <li>• deprecated on EM/MC74xx(9x30) devices</li> </ul>

<i>carrier_str</i>	Carrier String.
<i>priversion_str</i>	PRI Version String.
<i>priversion_str</i>	PRI Version String.
<i>cur_carr_name</i>	Current Carrier Name String.
<i>cur_carr_rev</i>	Current Carrier Revision String.
<i>Tlvresult</i>	Tlv Result.

## 8.847.2 Field Documentation

8.847.2.1 char unpack\_dms\_GetFirmwareInfo\_t::appversion\_str[85]

8.847.2.2 char unpack\_dms\_GetFirmwareInfo\_t::bootversion\_str[85]

8.847.2.3 char unpack\_dms\_GetFirmwareInfo\_t::carrier\_str[20]

8.847.2.4 char unpack\_dms\_GetFirmwareInfo\_t::cur\_carr\_name[17]

8.847.2.5 char unpack\_dms\_GetFirmwareInfo\_t::cur\_carr\_rev[13]

8.847.2.6 char unpack\_dms\_GetFirmwareInfo\_t::modelid\_str[20]

8.847.2.7 char unpack\_dms\_GetFirmwareInfo\_t::packageid\_str[85]

8.847.2.8 char unpack\_dms\_GetFirmwareInfo\_t::priversion\_str[16]

8.847.2.9 char unpack\_dms\_GetFirmwareInfo\_t::sku\_str[15]

8.847.2.10 uint16\_t unpack\_dms\_GetFirmwareInfo\_t::Tlvresult

## 8.848 unpack\_dms\_GetFirmwareRevision\_t Struct Reference

### Data Fields

- uint8\_t [amssSize](#)
- char [AMSSString](#) [255]
- char [PRIString](#) [255]
- uint16\_t [Tlvresult](#)

## 8.848.1 Field Documentation

8.848.1.1 uint8\_t unpack\_dms\_GetFirmwareRevision\_t::amssSize

8.848.1.2 char unpack\_dms\_GetFirmwareRevision\_t::AMSSString[255]

8.848.1.3 char unpack\_dms\_GetFirmwareRevision\_t::PRIString[255]

8.848.1.4 uint16\_t unpack\_dms\_GetFirmwareRevision\_t::Tlvresult

## 8.849 unpack\_dms\_GetFirmwareRevisions\_t Struct Reference

### Data Fields

- uint8\_t [amssSize](#)

- char [AMSSString](#) [255]
- uint8\_t [bootSize](#)
- char [BootString](#) [255]
- uint8\_t [priSize](#)
- char [PRIString](#) [255]
- uint16\_t [Tlvresult](#)

### 8.849.1 Detailed Description

#### Parameters

<i>amssstring</i>	AMSS revision string
<i>bootstring</i>	boot code revision string
<i>pristring</i>	PRI revision string

### 8.849.2 Field Documentation

- 8.849.2.1 uint8\_t [unpack\\_dms\\_GetFirmwareRevisions\\_t::amssSize](#)
- 8.849.2.2 char [unpack\\_dms\\_GetFirmwareRevisions\\_t::AMSSString](#)[255]
- 8.849.2.3 uint8\_t [unpack\\_dms\\_GetFirmwareRevisions\\_t::bootSize](#)
- 8.849.2.4 char [unpack\\_dms\\_GetFirmwareRevisions\\_t::BootString](#)[255]
- 8.849.2.5 uint8\_t [unpack\\_dms\\_GetFirmwareRevisions\\_t::priSize](#)
- 8.849.2.6 char [unpack\\_dms\\_GetFirmwareRevisions\\_t::PRIString](#)[255]
- 8.849.2.7 uint16\_t [unpack\\_dms\\_GetFirmwareRevisions\\_t::Tlvresult](#)

## 8.850 [unpack\\_dms\\_GetFSN\\_t](#) Struct Reference

#### Data Fields

- char [String](#) [255]
- uint16\_t [Tlvresult](#)

### 8.850.1 Field Documentation

- 8.850.1.1 char [unpack\\_dms\\_GetFSN\\_t::String](#)[255]
- 8.850.1.2 uint16\_t [unpack\\_dms\\_GetFSN\\_t::Tlvresult](#)

## 8.851 [unpack\\_dms\\_GetHardwareRevision\\_t](#) Struct Reference

#### Data Fields

- char [hwVer](#) [255]

### 8.851.1 Detailed Description

## Parameters

<i>hwVer</i>	hardware vesion
--------------	-----------------

### 8.851.2 Field Documentation

8.851.2.1 char unpack\_dms\_GetHardwareRevision\_t::hwVer[255]

## 8.852 unpack\_dms\_GetIMSI\_t Struct Reference

## Data Fields

- char [imsi](#) [255]
- uint16\_t [Tlvresult](#)

### 8.852.1 Field Documentation

8.852.1.1 char unpack\_dms\_GetIMSI\_t::imsi[255]

8.852.1.2 uint16\_t unpack\_dms\_GetIMSI\_t::Tlvresult

## 8.853 unpack\_dms\_GetModelID\_t Struct Reference

## Data Fields

- char [modelid](#) [255]
- uint16\_t [Tlvresult](#)

### 8.853.1 Detailed Description

## Parameters

<i>modelid</i>	device model id
----------------	-----------------

### 8.853.2 Field Documentation

8.853.2.1 char unpack\_dms\_GetModelID\_t::modelid[255]

8.853.2.2 uint16\_t unpack\_dms\_GetModelID\_t::Tlvresult

## 8.854 unpack\_dms\_GetNetworkTime\_t Struct Reference

## Data Fields

- uint16\_t [source](#)
- uint64\_t [timestamp](#)
- uint16\_t [Tlvresult](#)

### 8.854.1 Detailed Description

## Parameters

<i>source</i>	<ul style="list-style-type: none"> <li>Source of timestamp</li> <li>0 - 32 kHz device clock</li> <li>1 - CDMA network</li> <li>2 - cdma2000 1xEV-DO network</li> </ul>
<i>timestamp</i>	<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>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

## Note

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.

## 8.854.2 Field Documentation

8.854.2.1 `uint16_t unpack_dms_GetNetworkTime_t::source`

8.854.2.2 `uint64_t unpack_dms_GetNetworkTime_t::timestamp`

8.854.2.3 `uint16_t unpack_dms_GetNetworkTime_t::Tlvresult`

8.855 `unpack_dms_GetPower_t` Struct Reference

## Data Fields

- `uint32_t` [OperationMode](#)
- `uint32_t` [OfflineReason](#)
- `uint32_t` [HardwareControlledMode](#)
- `uint16_t` [Tlvresult](#)

## 8.855.1 Detailed Description

## Parameters

<i>OperationMode</i>	operating mode
<i>OfflineReason</i>	offline reason
<i>Hardware-ControlledMode</i>	hardware restricted mode

## 8.855.2 Field Documentation

8.855.2.1 `uint32_t unpack_dms_GetPower_t::HardwareControlledMode`

8.855.2.2 `uint32_t unpack_dms_GetPower_t::OfflineReason`

8.855.2.3 `uint32_t unpack_dms_GetPower_t::OperationMode`

8.855.2.4 uint16\_t unpack\_dms\_GetPower\_t::Tlvresult

## 8.856 unpack\_dms\_GetPRLVersion\_t Struct Reference

### Data Fields

- uint8\_t [u8PRLPreference](#)
- uint16\_t [u16PRLVersion](#)
- uint16\_t [Tlvresult](#)

### 8.856.1 Field Documentation

8.856.1.1 uint16\_t unpack\_dms\_GetPRLVersion\_t::Tlvresult

8.856.1.2 uint16\_t unpack\_dms\_GetPRLVersion\_t::u16PRLVersion

8.856.1.3 uint8\_t unpack\_dms\_GetPRLVersion\_t::u8PRLPreference

## 8.857 unpack\_dms\_GetSerialNumbers\_t Struct Reference

### Data Fields

- char [esn](#) [255]
- char [imei\\_no](#) [255]
- char [meid](#) [255]
- char [imeisv\\_svn](#) [255]

### 8.857.1 Detailed Description

#### Parameters

<i>esn</i>	Electronic Serial Number of the device
<i>imei_no</i>	International Mobile Equipment Identity of the device.
<i>meid</i>	Mobile Equipment Identifier of the device.
<i>imeisv_svn</i>	imei software version revision

### 8.857.2 Field Documentation

8.857.2.1 char unpack\_dms\_GetSerialNumbers\_t::esn[255]

8.857.2.2 char unpack\_dms\_GetSerialNumbers\_t::imei\_no[255]

8.857.2.3 char unpack\_dms\_GetSerialNumbers\_t::imeisv\_svn[255]

8.857.2.4 char unpack\_dms\_GetSerialNumbers\_t::meid[255]

## 8.858 unpack\_dms\_GetUSBComp\_t Struct Reference

### Data Fields

- uint8\_t [USBComp](#) [255]
- uint8\_t [NumSupUSBComps](#)

- uint8\_t [SupUSBComps](#)
- uint16\_t [Tlvresult](#)

### 8.858.1 Field Documentation

8.858.1.1 uint8\_t unpack\_dms\_GetUSBComp\_t::NumSupUSBComps

8.858.1.2 uint8\_t unpack\_dms\_GetUSBComp\_t::SupUSBComps

8.858.1.3 uint16\_t unpack\_dms\_GetUSBComp\_t::Tlvresult

8.858.1.4 uint8\_t unpack\_dms\_GetUSBComp\_t::USBComp[255]

## 8.859 unpack\_dms\_GetVoiceNumber\_t Struct Reference

### Data Fields

- uint8\_t [voiceNumberSize](#)
- char [VoiceNumber](#) [255]
- uint8\_t [minSize](#)
- char [MIN](#) [255]
- uint16\_t [Tlvresult](#)

### 8.859.1 Field Documentation

8.859.1.1 char unpack\_dms\_GetVoiceNumber\_t::MIN[255]

8.859.1.2 uint8\_t unpack\_dms\_GetVoiceNumber\_t::minSize

8.859.1.3 uint16\_t unpack\_dms\_GetVoiceNumber\_t::Tlvresult

8.859.1.4 char unpack\_dms\_GetVoiceNumber\_t::VoiceNumber[255]

8.859.1.5 uint8\_t unpack\_dms\_GetVoiceNumber\_t::voiceNumberSize

## 8.860 unpack\_dms\_SetCrashAction\_t Struct Reference

### Data Fields

- uint8\_t [notused](#)

### 8.860.1 Detailed Description

Modem response. Not used

### 8.860.2 Field Documentation

8.860.2.1 uint8\_t unpack\_dms\_SetCrashAction\_t::notused

## 8.861 unpack\_dms\_SetCustFeature\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.861.1 Field Documentation

8.861.1.1 uint16\_t unpack\_dms\_SetCustFeature\_t::Tlvresult

## 8.862 unpack\_dms\_SetCustFeaturesV2\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.862.1 Detailed Description

This structure contains customization settings set to modem unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
------------------	---

### 8.862.2 Field Documentation

8.862.2.1 uint16\_t unpack\_dms\_SetCustFeaturesV2\_t::Tlvresult

## 8.863 unpack\_dms\_SetEventReport\_ind\_t Struct Reference

## Data Fields

- [dms\\_ActivationStatusTlv](#) ActivationStatusTlv
- [dms\\_OperatingModeTlv](#) OperatingModeTlv
- uint16\_t [Tlvresult](#)

### 8.863.1 Detailed Description

DMS Event Report indication structure

#### Parameters

<i>ActivationStatus-Tlv</i>	<ul style="list-style-type: none"><li>• See <a href="#">dms_ActivationStatusTlv</a></li></ul>
<i>OperatingMode-Tlv</i>	<ul style="list-style-type: none"><li>• See <a href="#">dms_OperatingModeTlv</a></li></ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>

### 8.863.2 Field Documentation

8.863.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.863.2.2 `dms_OperatingModeTlv unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.863.2.3 `uint16_t unpack_dms_SetEventReport_ind_t::Tlvresult`

## 8.864 `unpack_dms_SetEventReport_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.864.1 Field Documentation

8.864.1.1 `uint16_t unpack_dms_SetEventReport_t::Tlvresult`

## 8.865 `unpack_dms_SetFirmwarePreference_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.865.1 Field Documentation

8.865.1.1 `uint16_t unpack_dms_SetFirmwarePreference_t::Tlvresult`

## 8.866 `unpack_dms_SetPower_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.866.1 Field Documentation

8.866.1.1 `uint16_t unpack_dms_SetPower_t::Tlvresult`

## 8.867 `unpack_dms_SetUSBComp_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.867.1 Field Documentation

8.867.1.1 `uint16_t unpack_dms_SetUSBComp_t::Tlvresult`

## 8.868 `unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t` Struct Reference

## Data Fields

- uint8\_t [type](#)
- uint8\_t [source](#)
- uint16\_t [Tlvresult](#)

## 8.868.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

## Parameters

<i>OUT]</i>	type[OUT] <ul style="list-style-type: none"> <li>• type of reset or power down, possible values listed below:               <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - warm</li> <li>– 2 - hard</li> <li>– 3 - crash</li> <li>– 4 - power down</li> </ul> </li> </ul>
<i>OUT]</i>	source[OUT] <ul style="list-style-type: none"> <li>• entity which initiated the reset or power down, possible values listed below:               <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - user requested ( AT!RESET, AT!BOOTHOLD, FW/PRI download – including host-initiated image switching)</li> <li>– 2 - hardware switch (W_DISABLE)</li> <li>– 3 - temperature critical</li> <li>– 4 - voltage critical</li> <li>– 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset)</li> <li>– 6 - LWM2M (Light Weight M2M client (internal process for LWM2M))</li> <li>– 7 - OMA-DM</li> <li>– 8 - FOTA</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

## 8.868.2 Field Documentation

8.868.2.1 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::source

8.868.2.2 uint16\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::Tlvresult

8.868.2.3 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t::type

## 8.869 unpack\_dms\_SLQSDmsSwiGetResetInfo\_t Struct Reference

## Data Fields

- uint8\_t [type](#)
- uint8\_t [source](#)
- uint16\_t [Tlvresult](#)

### 8.869.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

#### Parameters

<i>OUT]</i>	type[OUT] <ul style="list-style-type: none"> <li>• type of reset or power down, possible values listed below:               <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - warm</li> <li>– 2 - hard</li> <li>– 3 - crash</li> <li>– 4 - power down</li> </ul> </li> </ul>
<i>OUT]</i>	source[OUT] <ul style="list-style-type: none"> <li>• entity which initiated the reset or power down, possible values listed below:               <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - user requested ( AT!RESET, AT!BOOTHOLD, FW/PRI download – including host-initiated image switching)</li> <li>– 2 - hardware switch (W_DISABLE)</li> <li>– 3 - temperature critical</li> <li>– 4 - voltage critical</li> <li>– 5 - configuration update (SIM-based image switching, RMA reset, NVUPs which request a reset)</li> <li>– 6 - LWM2M (Light Weight M2M client (internal process for LWM2M))</li> <li>– 7 - OMA-DM</li> <li>– 8 - FOTA</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

### 8.869.2 Field Documentation

8.869.2.1 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::source

8.869.2.2 uint16\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::Tlvresult

8.869.2.3 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::type

## 8.870 unpack\_dms\_SLQSDmsSwiIndicationRegister\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.870.1 Detailed Description

This structure contains set registration state for different indication unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>Unpack Result</li></ul>
------------------	---

## 8.870.2 Field Documentation

8.870.2.1 uint16\_t unpack\_dms\_SLQSDmsSwilIndicationRegister\_t::Tlvresult

## 8.871 unpack\_dms\_SLQSGetBandCapability\_t Struct Reference

## Data Fields

- uint64\_t [bandCapability](#)
- int [is\\_LteBandCapability\\_Available](#)
- uint64\_t [LteBandCapability](#)
- int [is\\_TdsBandCapability\\_Available](#)
- uint64\_t [TdsBandCapability](#)

## 8.871.1 Detailed Description

This structure contains the Band Capabilities response.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

Parameters

---

<i>bandCapability[OUT]</i>	<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> <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> <li>• Bit 56 - Band class 16</li> <li>• Bit 57 - Band class 17</li> <li>• Bit 58 - Band class 18</li> <li>• Bit 59 - Band class 19</li> </ul>
----------------------------	---

<i>LteBand-Capability[OUT]</i>	<p>Bitmask of LTE bands supported by the device</p> <ul style="list-style-type: none"> <li>• Bit 0 - LTE EUTRAN Band 1 UL:1920-1980; DL: 2110-2170</li> <li>• Bit 1 - LTE EUTRAN Band 2 UL:1850-1910; DL: 1930-1990</li> <li>• Bit 2 - LTE EUTRAN Band 3 UL:1710-1785; DL: 1805-1880</li> <li>• Bit 3 - LTE EUTRAN Band 4 UL:1710-1755; DL: 2110-2155</li> <li>• Bit 4 - LTE EUTRAN Band 5 UL: 824-849; DL: 869-894</li> <li>• Bit 5 - LTE EUTRAN Band 6 UL: 830-840; DL: 875-885</li> <li>• Bit 6 - LTE EUTRAN Band 7 UL:2500-2570; DL: 2620-2690</li> <li>• Bit 7 - LTE EUTRAN Band 8 UL: 880-915; DL: 925-960</li> <li>• Bit 8 - LTE EUTRAN Band 9 UL:1749.9-1784.9; DL: 1844.9-1879.9</li> <li>• Bit 9 - LTE EUTRAN Band 10 UL:1710-1770; DL: 2110-2170</li> <li>• Bit 10 - LTE EUTRAN Band 11 UL:1427.9-1452.9; DL: 1475.9-1500.9</li> <li>• Bit 11 - LTE EUTRAN Band 12 UL:698-716; DL: 728-746</li> <li>• Bit 12 - LTE EUTRAN Band 13 UL: 777-787; DL: 746-756</li> <li>• Bit 13 - LTE EUTRAN Band 14 UL: 788-798; DL: 758-768</li> <li>• Bits 14 and 15 - Reserved</li> <li>• Bit 16 - LTE EUTRAN Band 17 UL: 704-716; DL: 734-746</li> <li>• Bit 17 - LTE EUTRAN Band 18 UL: 815-830; DL: 860-875</li> <li>• Bit 18 - LTE EUTRAN Band 19 UL: 830-845; DL: 875-890</li> <li>• Bit 19 - LTE EUTRAN Band 20 UL: 832-862; DL: 791-821</li> <li>• Bit 20 - LTE EUTRAN Band 21 UL: 1447.9-1462.9; DL: 1495.9-1510.9</li> <li>• Bit 21 - Reserved</li> <li>• Bit 22 - LTE EUTRAN Band 23 UL: 2000-2020; DL: 2180-2200</li> <li>• Bit 23 - LTE EUTRAN Band 24 UL: 1626.5-1660.5; DL: 1525-1559</li> <li>• Bit 24 - LTE EUTRAN Band 25 UL: 1850-1915; DL: 1930-1995</li> <li>• Bit 25 - LTE EUTRAN Band 26 UL: 814-849; DL: 859-894</li> <li>• Bit 26 - Reserved</li> <li>• Bit 27 - LTE EUTRAN Band 28 UL: 703-748; DL: 758-803</li> <li>• Bit 28 - LTE EUTRAN Band 29 UL: 1850-1910 or 1710-1755; DL: 716-728</li> <li>• Bits 29 through 31 - Reserved</li> <li>• Bit 32 - LTE EUTRAN Band 33 UL: 1900-1920; DL: 1900-1920</li> <li>• Bit 33 - LTE EUTRAN Band 34 UL: 2010-2025; DL: 2010-2025</li> <li>• Bit 34 - LTE EUTRAN Band 35 UL: 1850-1910; DL: 1850-1910</li> <li>• Bit 35 - LTE EUTRAN Band 36 UL: 1930-1990; DL: 1930-1990</li> <li>• Bit 36 - LTE EUTRAN Band 37 UL: 1910-1930; DL: 1910-1930</li> <li>• Bit 37 - LTE EUTRAN Band 38 UL: 2570-2620; DL: 2570-2620</li> <li>• Bit 38 - LTE EUTRAN Band 39 UL: 1880-1920; DL: 1880-1920</li> <li>• Bit 39 - LTE EUTRAN Band 40 UL: 2300-2400; DL: 2300-2400</li> <li>• Bit 40 - LTE EUTRAN Band 41 UL: 2496-2690; DL: 2496-2690</li> <li>• Bit 41 - LTE EUTRAN Band 42 UL: 3400-3600; DL: 3400-3600</li> <li>• Bit 42 - LTE EUTRAN Band 43 UL: 3600-3800; DL: 3600-3800</li> <li>• Bits 43 through 64 - Reserved</li> </ul>
--------------------------------	--

<i>TdsBand-Capability</i> [OUT]	Bitmask of TDS bands supported by the device. <ul style="list-style-type: none"> <li>• Bit 0 - TDS Band A 1900 to 1920 MHz, 2010 to 2020 MHz</li> <li>• Bit 1 - TDS Band B 1850 to 1910 MHz, 1930 to 1990 MHz</li> <li>• Bit 2 - TDS Band C 1910 to 1930 MHz</li> <li>• Bit 3 - TDS Band D 2570 to 2620 MHz</li> <li>• Bit 4 - TDS Band E 2300 to 2400 MHz</li> <li>• Bit 5 - TDS Band F 1880 to 1920 MHz</li> </ul>
---------------------------------	--

## 8.871.2 Field Documentation

8.871.2.1 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::bandCapability

8.871.2.2 int unpack\_dms\_SLQSGetBandCapability\_t::is\_LteBandCapability\_Available

8.871.2.3 int unpack\_dms\_SLQSGetBandCapability\_t::is\_TdsBandCapability\_Available

8.871.2.4 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::LteBandCapability

8.871.2.5 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::TdsBandCapability

## 8.872 unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.872.1 Detailed Description

This structure contains Clear Dying GASP unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
------------------	---

## 8.872.2 Field Documentation

8.872.2.1 uint16\_t unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t::Tlvresult

## 8.873 unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t Struct Reference

### Data Fields

- [packgetDyingGaspCfg](#) \* [pGetDyingGaspCfg](#)
- uint16\_t [Tlvresult](#)

### 8.873.1 Detailed Description

This structure contains Get Dying GASP Config unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
------------------	---

### 8.873.2 Field Documentation

8.873.2.1 `packgetDyingGaspCfg*` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.873.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

## 8.874 unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t Struct Reference

#### Data Fields

- `packgetDyingGaspStatistics*` `pGetDyingGaspStatistics`
- `uint16_t` `Tlvresult`

### 8.874.1 Detailed Description

This structure contains Get Dying GASP Statistics.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
------------------	---

### 8.874.2 Field Documentation

8.874.2.1 `packgetDyingGaspStatistics*` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.874.2.2 `uint16_t` `unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

## 8.875 unpack\_dms\_SLQSSwiGetFirmwareCurr\_t Struct Reference

#### Data Fields

- `uint8_t` `numEntries`
- `image_info_t*` `pCurrImgInfo`
- `char` `priver` [16]
- `char` `pkgver` [16]
- `char` `fwvers` [16]
- `char` `carrier` [16]

### 8.875.1 Detailed Description

## 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 pCurrImgInfo 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>

## 8.875.2 Field Documentation

8.875.2.1 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::carrier[16]

8.875.2.2 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::fwvers[16]

8.875.2.3 uint8\_t unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::numEntries

8.875.2.4 image\_info\_t\* unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::pCurrImgInfo

8.875.2.5 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::pkgver[16]

8.875.2.6 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::priver[16]

## 8.876 unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t Struct Reference

## Data Fields

- uint32\_t [ResCode](#)
- uint8\_t [imgType](#)
- uint32\_t [refData](#)
- uint8\_t [refString](#) [15]
- uint8\_t [logString](#) [255]
- uint16\_t [Tlvresult](#)

## 8.876.1 Detailed Description

This structure is used to store Firmware Update Status

## Parameters

<i>ResCode</i>	<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>0x40000nnn - 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>imgType</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Firmware image type that failed the update</li> </ul>
<i>refData</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Failed image reference data</li> <li>This is normally the offset of the image that caused the failure</li> </ul>
<i>refString</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>
<i>logString</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Failed image reference string. This is normally the partition name of the image that caused the failure if applicable.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>

## 8.876.2 Field Documentation

8.876.2.1 uint8\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::imgType

8.876.2.2 uint8\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::logString[255]

8.876.2.3 uint32\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::refData

8.876.2.4 uint8\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::refString[15]

8.876.2.5 uint32\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::ResCode

8.876.2.6 uint16\_t unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t::Tlvresult

## 8.877 unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

#### 8.877.1 Detailed Description

This structure contains set Dying GASP Config unpack

##### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
------------------	---

#### 8.877.2 Field Documentation

8.877.2.1 uint16\_t unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t::Tlvresult

## 8.878 unpack\_dms\_UIMGetlCCID\_t Struct Reference

### Data Fields

- uint8\_t [stringSize](#)
- uint8\_t [String](#) [255]
- uint16\_t [Tlvresult](#)

#### 8.878.1 Detailed Description

This structure contains Get ICCID pack

##### Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> <li>• Size of String.</li> </ul>
<i>String</i>	<ul style="list-style-type: none"> <li>• ICCID String.</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack result.</li> </ul>

#### 8.878.2 Field Documentation

8.878.2.1 uint8\_t unpack\_dms\_UIMGetlCCID\_t::String[255]

8.878.2.2 uint8\_t unpack\_dms\_UIMGetlCCID\_t::stringSize

8.878.2.3 uint16\_t unpack\_dms\_UIMGetICCID\_t::Tlvresult

## 8.879 unpack\_fms\_GetImagesPreference\_t Struct Reference

### Data Fields

- uint32\_t [ImageListSize](#)
- FMSPrefImageList \* [pImageList](#)
- uint16\_t [Tlvresult](#)

### 8.879.1 Detailed Description

This structure contains the Get Image Preference information unpack

#### 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="#">FMSImageElement</a></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

### 8.879.2 Field Documentation

8.879.2.1 uint32\_t unpack\_fms\_GetImagesPreference\_t::ImageListSize

8.879.2.2 FMSPrefImageList\* unpack\_fms\_GetImagesPreference\_t::pImageList

8.879.2.3 uint16\_t unpack\_fms\_GetImagesPreference\_t::Tlvresult

## 8.880 unpack\_fms\_GetStoredImages\_t Struct Reference

### Data Fields

- uint32\_t [imagelistSize](#)
- [FMSImageList](#) [imageList](#)
- uint16\_t [Tlvresult](#)

### 8.880.1 Detailed Description

This structure contains the Get Stored Images unpack

#### Parameters

<i>listSize</i>	<ul style="list-style-type: none"> <li>• The number of elements in the image list</li> </ul>
-----------------	--

<i>imageList</i>	<ul style="list-style-type: none"> <li>• Array of Image entries with size provided by previous field</li> <li>• See <a href="#">FMSImageElement</a></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

## 8.880.2 Field Documentation

8.880.2.1 **FMSImageList** `unpack_fms_GetStoredImages_t::imageList`

8.880.2.2 `uint32_t` `unpack_fms_GetStoredImages_t::imagelistSize`

8.880.2.3 `uint16_t` `unpack_fms_GetStoredImages_t::Tlvresult`

## 8.881 `unpack_fms_SetImagesPreference_t` Struct Reference

### Data Fields

- `uint32_t` [ImageTypesSize](#)
- `uint8_t` [ImageTypes](#) [255]
- `uint16_t` [Tlvresult](#)

### 8.881.1 Detailed Description

This structure contains the Set Images Preference unpack

#### Parameters

<i>ImageTypesSize</i>	<ul style="list-style-type: none"> <li>• Image Type Size</li> </ul>
<i>ImageTypes</i>	<ul style="list-style-type: none"> <li>• Image Type</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

## 8.881.2 Field Documentation

8.881.2.1 `uint8_t` `unpack_fms_SetImagesPreference_t::ImageTypes[255]`

8.881.2.2 `uint32_t` `unpack_fms_SetImagesPreference_t::ImageTypesSize`

8.881.2.3 `uint16_t` `unpack_fms_SetImagesPreference_t::Tlvresult`

## 8.882 `unpack_loc_BestAvailPos_Ind_t` Struct Reference

## Data Fields

- uint32\_t [status](#)
- uint32\_t \* [pXid](#)
- uint64\_t \* [pLatitude](#)
- uint64\_t \* [pLongitude](#)
- uint32\_t \* [pHorUncCircular](#)
- uint32\_t \* [pAltitudeWrtEllipsoid](#)
- uint32\_t \* [pVertUnc](#)
- uint64\_t \* [pTimestampUtc](#)
- uint32\_t \* [pTimeUnc](#)
- uint32\_t \* [pHorUncEllipseSemiMinor](#)
- uint32\_t \* [pHorUncEllipseSemiMajor](#)
- uint32\_t \* [pHorUncEllipseOrientAzimuth](#)
- uint8\_t \* [pHorCirConf](#)
- uint8\_t \* [pHorEllpConf](#)
- uint32\_t \* [pHorReliability](#)
- uint32\_t \* [pSpeedHorizontal](#)
- uint32\_t \* [pSpeedUnc](#)
- uint32\_t \* [pAltitudeWrtMeanSeaLevel](#)
- uint8\_t \* [pVertConfidence](#)
- uint32\_t \* [pVertReliability](#)
- uint32\_t \* [pSpeedVertical](#)
- uint32\_t \* [pSpeedVerticalUnc](#)
- uint32\_t \* [pHeading](#)
- uint32\_t \* [pHeadingUnc](#)
- uint32\_t \* [pMagneticDeviation](#)
- uint32\_t \* [pTechnologyMask](#)
- [loc\\_precisionDilution](#) \* [pPrecisionDilution](#)
- [loc\\_gpsTime](#) \* [pGpsTime](#)
- uint32\_t \* [pTimeSrc](#)
- [loc\\_sensorDataUsage](#) \* [pSensorDataUsage](#)
- [loc\\_svUsedforFix](#) \* [pSvUsedforFix](#)
- uint16\_t [Tlvresult](#)

### 8.882.1 Detailed Description

This structure contains Best Available Position

## Parameters

<i>status</i>	<ul style="list-style-type: none"> <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>
<i>xid</i>	Transaction ID that was specified in the Get Best Available Position request.
<i>pLatitude</i>	<ul style="list-style-type: none"> <li>Latitude (specified in WGS84 datum)</li> <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>Longitude (specified in WGS84 datum)</li> <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>pAltitudeWrt-Ellipsoid</i>	<ul style="list-style-type: none"> <li>Altitude With Respect to WGS84 Ellipsoid.</li> <li>Units - Meters</li> <li>Range -500 to 15883</li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>Vertical uncertainty.</li> <li>Units - Meters</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>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</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>pHorCirConf</i>	<ul style="list-style-type: none"> <li>• Horizontal circular uncertainty confidence</li> <li>• Units: Precent</li> <li>• Range: 0 to 99</li> </ul>
<i>pHorEllpConf</i>	<ul style="list-style-type: none"> <li>• Horizontal elliptical uncertainty confidence</li> <li>• Units: Precent</li> <li>• Range: 0 to 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-MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Sea Level.</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>pSpeedVertical-Unc</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="#">loc_precisionDilution</a> for more information</li> </ul>
<i>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gpsTime</a> for more information</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> <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> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </li> </ul>
<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_sensorDataUsage</a> for more information</li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_svUsedforFix</a> for more information</li> </ul>

## 8.882.2 Field Documentation

8.882.2.1 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pAltitudeWrtEllipsoid

8.882.2.2 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pAltitudeWrtMeanSeaLevel

8.882.2.3 loc\_gpsTime\* unpack\_loc\_BestAvailPos\_Ind\_t::pGpsTime

8.882.2.4 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHeading

8.882.2.5 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHeadingUnc

8.882.2.6 uint8\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorCirConf

8.882.2.7 uint8\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorEllpConf

8.882.2.8 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorReliability

8.882.2.9 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorUncCircular

8.882.2.10 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorUncEllipseOrientAzimuth

8.882.2.11 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pHorUncEllipseSemiMajor

8.882.2.12 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor`

8.882.2.13 `uint64_t*` `unpack_loc_BestAvailPos_Ind_t::pLatitude`

8.882.2.14 `uint64_t*` `unpack_loc_BestAvailPos_Ind_t::pLongitude`

8.882.2.15 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation`

8.882.2.16 `loc_precisionDilution*` `unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution`

8.882.2.17 `loc_sensorDataUsage*` `unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage`

8.882.2.18 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal`

8.882.2.19 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedUnc`

8.882.2.20 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedVertical`

8.882.2.21 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc`

8.882.2.22 `loc_svUsedforFix*` `unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix`

8.882.2.23 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pTechnologyMask`

8.882.2.24 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pTimeSrc`

8.882.2.25 `uint64_t*` `unpack_loc_BestAvailPos_Ind_t::pTimestampUtc`

8.882.2.26 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pTimeUnc`

8.882.2.27 `uint8_t*` `unpack_loc_BestAvailPos_Ind_t::pVertConfidence`

8.882.2.28 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pVertReliability`

8.882.2.29 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pVertUnc`

8.882.2.30 `uint32_t*` `unpack_loc_BestAvailPos_Ind_t::pXid`

8.882.2.31 `uint32_t` `unpack_loc_BestAvailPos_Ind_t::status`

8.882.2.32 `uint16_t` `unpack_loc_BestAvailPos_Ind_t::Tlvresult`

## 8.883 `unpack_loc_Delete_Assist_Data_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.883.1 Detailed Description

This structure contains LOC delete assist data unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>Unpack result.</li></ul>
------------------	--

## 8.883.2 Field Documentation

8.883.2.1 uint16\_t unpack\_loc\_Delete\_Assist\_Data\_t::Tlvresult

## 8.884 unpack\_loc\_EngineState\_Ind\_t Struct Reference

## Data Fields

- uint32\_t [engineState](#)
- uint16\_t [Tlvresult](#)

## 8.884.1 Detailed Description

This structure contains LOC Engine State field.

## Parameters

<i>engineState</i>	<ul style="list-style-type: none"><li>Location engine state.</li><li>Valid values<ul style="list-style-type: none"><li>1 - Location engine is on</li><li>2 - Location engine is off</li></ul></li></ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"><li>unpack result</li></ul>

## 8.884.2 Field Documentation

8.884.2.1 uint32\_t unpack\_loc\_EngineState\_Ind\_t::engineState

8.884.2.2 uint16\_t unpack\_loc\_EngineState\_Ind\_t::Tlvresult

## 8.885 unpack\_loc\_EventRegister\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

## 8.885.1 Detailed Description

This structure contains Event Register unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>Unpack result.</li></ul>
------------------	--

## 8.885.2 Field Documentation

8.885.2.1 uint16\_t unpack\_loc\_EventRegister\_t::Tlvresult

## 8.886 unpack\_loc\_PositionRpt\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [sessionStatus](#)
- uint8\_t [sessionId](#)
- uint64\_t \* [pLatitude](#)
- uint64\_t \* [pLongitude](#)
- uint32\_t \* [pHorUncCircular](#)
- uint32\_t \* [pHorUncEllipseSemiMinor](#)
- uint32\_t \* [pHorUncEllipseSemiMajor](#)
- uint32\_t \* [pHorUncEllipseOrientAzimuth](#)
- uint8\_t \* [pHorConfidence](#)
- uint32\_t \* [pHorReliability](#)
- uint32\_t \* [pSpeedHorizontal](#)
- uint32\_t \* [pSpeedUnc](#)
- uint32\_t \* [pAltitudeWrtEllipsoid](#)
- uint32\_t \* [pAltitudeWrtMeanSeaLevel](#)
- uint32\_t \* [pVertUnc](#)
- uint8\_t \* [pVertConfidence](#)
- uint32\_t \* [pVertReliability](#)
- uint32\_t \* [pSpeedVertical](#)
- uint32\_t \* [pHeading](#)
- uint32\_t \* [pHeadingUnc](#)
- uint32\_t \* [pMagneticDeviation](#)
- uint32\_t \* [pTechnologyMask](#)
- [loc\\_precisionDilution](#) \* [pPrecisionDilution](#)
- uint64\_t \* [pTimestampUtc](#)
- uint8\_t \* [pLeapSeconds](#)
- [loc\\_gpsTime](#) \* [pGpsTime](#)
- uint32\_t \* [pTimeUnc](#)
- uint32\_t \* [pTimeSrc](#)
- [loc\\_sensorDataUsage](#) \* [pSensorDataUsage](#)
- uint32\_t \* [pFixId](#)
- [loc\\_svUsedforFix](#) \* [pSvUsedforFix](#)
- uint8\_t \* [pAltitudeAssumed](#)
- uint16\_t [Tlvresult](#)

### 8.886.1 Detailed Description

This structure contains Event Position Report Indication unpack

## 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>• Latitude (specified in WGS84 datum)</li> <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>• Longitude (specified in WGS84 datum)</li> <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 WGS84 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="#">loc_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="#">loc_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> <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> <li>– 6 - Both time of the week and the GPS week number are known.</li> <li>– 7 - Time is set by the position engine after the fix is obtained</li> <li>– 8 - Time is set by the position engine after performing SFT, this is done when the clock time uncertainty is large.</li> <li>– 9 - Time is set after decoding GLO satellites.</li> <li>– 10- Time is set after transforming the GPS to GLO time</li> <li>– 11- Time is set by the sleep time tag provided by the WCDMA network.</li> <li>– 12- Time is set by the sleep time tag provided by the GSM network</li> <li>– 13- Source of the time is unknown</li> <li>– 14- Time is derived from the system clock (better known as the slow clock); GNSS time is maintained irrespective of the GNSS receiver state</li> <li>– 15- Time is set after decoding QZSS satellites.</li> <li>– 16- Time is set after decoding BDS satellites.</li> </ul> </li> </ul>
<i>-pSensorData-Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_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="#">loc_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

## 8.886.2 Field Documentation

8.886.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`

8.886.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`

8.886.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`

8.886.2.4 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`

8.886.2.5 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`

- 8.886.2.6 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHeading
- 8.886.2.7 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHeadingUnc
- 8.886.2.8 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorConfidence
- 8.886.2.9 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorReliability
- 8.886.2.10 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncCircular
- 8.886.2.11 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseOrientAzimuth
- 8.886.2.12 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseSemiMajor
- 8.886.2.13 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pHorUncEllipseSemiMinor
- 8.886.2.14 uint64\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLatitude
- 8.886.2.15 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLeapSeconds
- 8.886.2.16 uint64\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pLongitude
- 8.886.2.17 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pMagneticDeviation
- 8.886.2.18 loc\_precisionDilution\* unpack\_loc\_PositionRpt\_Ind\_t::pPrecisionDilution
- 8.886.2.19 loc\_sensorDataUsage\* unpack\_loc\_PositionRpt\_Ind\_t::pSensorDataUsage
- 8.886.2.20 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pSpeedHorizontal
- 8.886.2.21 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pSpeedUnc
- 8.886.2.22 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pSpeedVertical
- 8.886.2.23 loc\_svUsedforFix\* unpack\_loc\_PositionRpt\_Ind\_t::pSvUsedforFix
- 8.886.2.24 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pTechnologyMask
- 8.886.2.25 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pTimeSrc
- 8.886.2.26 uint64\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pTimestampUtc
- 8.886.2.27 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pTimeUnc
- 8.886.2.28 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertConfidence
- 8.886.2.29 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertReliability
- 8.886.2.30 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertUnc
- 8.886.2.31 uint8\_t unpack\_loc\_PositionRpt\_Ind\_t::sessionId
- 8.886.2.32 uint32\_t unpack\_loc\_PositionRpt\_Ind\_t::sessionIdStatus
- 8.886.2.33 uint16\_t unpack\_loc\_PositionRpt\_Ind\_t::Tlvresult

## 8.887 unpack\_loc\_SetExtPowerConfig\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [status](#)
- uint16\_t [Tlvresult](#)

### 8.887.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Valid 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>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.887.2 Field Documentation

8.887.2.1 uint32\_t unpack\_loc\_SetExtPowerConfig\_Ind\_t::status

8.887.2.2 uint16\_t unpack\_loc\_SetExtPowerConfig\_Ind\_t::Tlvresult

## 8.888 unpack\_loc\_SetExtPowerState\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.888.1 Detailed Description

This structure contains Set Ext Power State unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> </ul>
------------------	--

## 8.888.2 Field Documentation

8.888.2.1 uint16\_t unpack\_loc\_SetExtPowerState\_t::Tlvresult

## 8.889 unpack\_loc\_SetOperationMode\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.889.1 Detailed Description

This structure contains Set Operation Mode unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack result.</li></ul>
------------------	--

## 8.889.2 Field Documentation

8.889.2.1 uint16\_t unpack\_loc\_SetOperationMode\_t::Tlvresult

## 8.890 unpack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.890.1 Detailed Description

This structure contains Set Operation Mode unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack result.</li></ul>
------------------	--

## 8.890.2 Field Documentation

8.890.2.1 uint16\_t unpack\_loc\_SLQSLOCGetBestAvailPos\_t::Tlvresult

## 8.891 unpack\_loc\_Start\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.891.1 Detailed Description

This structure contains Start LOC unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result.</li> </ul>
------------------	--

### 8.891.2 Field Documentation

8.891.2.1 `uint16_t unpack_loc_Start_t::Tlvresult`

## 8.892 unpack\_loc\_Stop\_t Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)

### 8.892.1 Detailed Description

This structure contains Stop LOC unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result.</li> </ul>
------------------	--

### 8.892.2 Field Documentation

8.892.2.1 `uint16_t unpack_loc_Stop_t::Tlvresult`

## 8.893 unpack\_nas\_GetCDMANetworkParameters\_t Struct Reference

### Data Fields

- `uint8_t` [SCI](#)
- `uint8_t` [SCM](#)
- `uint8_t` [RegHomeSID](#)
- `uint8_t` [RegForeignSID](#)
- `uint8_t` [RegForeignNID](#)
- `uint8_t` [ForceRev0](#)
- `uint8_t` [CustomSCP](#)
- `uint32_t` [Protocol](#)
- `uint32_t` [Broadcast](#)
- `uint32_t` [Application](#)
- `uint32_t` [Roaming](#)

### 8.893.1 Detailed Description

## Parameters

<i>SCI</i>	slot cycle index
<i>SCM</i>	station class mark
<i>RegHomeSID</i>	register on home sid
<i>RegForeignSID</i>	register on foreign sid
<i>RegForeignNID</i>	register on foreign nid
<i>ForceRev0</i>	force header revision
<i>CustomSCP</i>	custom SCP
<i>Protocol</i>	protocol
<i>Booadcast</i>	broadcast
<i>Application</i>	application
<i>Roaming</i>	roaming

## 8.893.2 Field Documentation

8.893.2.1 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Application

8.893.2.2 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Broadcast

8.893.2.3 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::CustomSCP

8.893.2.4 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::ForceRev0

8.893.2.5 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Protocol

8.893.2.6 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignNID

8.893.2.7 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignSID

8.893.2.8 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegHomeSID

8.893.2.9 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Roaming

8.893.2.10 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCI

8.893.2.11 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCM

## 8.894 unpack\_nas\_GetHomeNetwork\_t Struct Reference

## Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- char [name](#) [255]
- uint16\_t [sid](#)
- uint16\_t [nid](#)

## 8.894.1 Detailed Description

## Parameters

<i>mcc</i>	mobile country code
<i>mnc</i>	mobile network code
<i>name</i>	network name or description
<i>sid</i>	home network system id only applies to cdma2000

<i>nid</i>	home network id Only applies to cdma2000
------------	--

## 8.894.2 Field Documentation

8.894.2.1 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mcc

8.894.2.2 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mnc

8.894.2.3 char unpack\_nas\_GetHomeNetwork\_t::name[255]

8.894.2.4 uint16\_t unpack\_nas\_GetHomeNetwork\_t::nid

8.894.2.5 uint16\_t unpack\_nas\_GetHomeNetwork\_t::sid

## 8.895 unpack\_nas\_GetNetworkPreference\_t Struct Reference

### Data Fields

- uint32\_t [ActiveTechPref](#)
- uint32\_t [Duration](#)
- uint32\_t [PersistentTechPref](#)
- uint16\_t [Tlvresult](#)

### 8.895.1 Detailed Description

#### Parameters

<i>TechnologyPref[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>Duration[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>Persistent-TechnologyPref[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>

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>
------------------	---

## 8.895.2 Field Documentation

8.895.2.1 uint32\_t unpack\_nas\_GetNetworkPreference\_t::ActiveTechPref

8.895.2.2 uint32\_t unpack\_nas\_GetNetworkPreference\_t::Duration

8.895.2.3 uint32\_t unpack\_nas\_GetNetworkPreference\_t::PersistentTechPref

8.895.2.4 uint16\_t unpack\_nas\_GetNetworkPreference\_t::Tlvresult

## 8.896 unpack\_nas\_GetRFInfo\_t Struct Reference

### Data Fields

- uint8\_t [instancesSize](#)
- [RFBandInfoElements](#) [RFBandInfoElements](#) [255]

### 8.896.1 Detailed Description

#### Parameters

<i>instancesSize</i>	number of elements in RF info instances array.
<a href="#">RFBandInfo-Elements</a>	RF info instances array

## 8.896.2 Field Documentation

8.896.2.1 uint8\_t unpack\_nas\_GetRFInfo\_t::instancesSize

8.896.2.2 [RFBandInfoElements](#) unpack\_nas\_GetRFInfo\_t::RFBandInfoElements[255]

## 8.897 unpack\_nas\_GetServingNetwork\_t Struct Reference

### Data Fields

- uint32\_t [RegistrationState](#)
- uint32\_t [CSDomain](#)
- uint32\_t [PSDomain](#)
- uint32\_t [RAN](#)
- uint8\_t [RadiolfacesSize](#)
- uint8\_t [Radiolfaces](#) [255]
- uint32\_t [Roaming](#)
- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [nameSize](#)
- uint8\_t [Name](#) [255]
- uint8\_t [DataCapsLen](#)
- uint8\_t [DataCaps](#) [255]

### 8.897.1 Detailed Description

#### Parameters

<i>Registration-State</i>	registration state
<i>CSDomain</i>	CS domain
<i>PSDomain</i>	PS domain
<i>RAN</i>	radio access network
<i>RadiolfacesSize</i>	radio interface size
<i>Radiolfaces</i>	radio interface list
<i>Roaming</i>	romaing indicator
<i>MCC</i>	Mobile country code
<i>MNC</i>	Mobile network code
<i>nameSize</i>	network name size
<i>Name</i>	network name
<i>DataCapsLen</i>	data capabilities len
<i>DataCap</i>	data capabilities

### 8.897.2 Field Documentation

8.897.2.1 uint32\_t unpack\_nas\_GetServingNetwork\_t::CSDomain

8.897.2.2 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCaps[255]

8.897.2.3 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCapsLen

8.897.2.4 uint16\_t unpack\_nas\_GetServingNetwork\_t::MCC

8.897.2.5 uint16\_t unpack\_nas\_GetServingNetwork\_t::MNC

8.897.2.6 uint8\_t unpack\_nas\_GetServingNetwork\_t::Name[255]

8.897.2.7 uint8\_t unpack\_nas\_GetServingNetwork\_t::nameSize

8.897.2.8 uint32\_t unpack\_nas\_GetServingNetwork\_t::PSDomain

8.897.2.9 uint8\_t unpack\_nas\_GetServingNetwork\_t::Radiolfaces[255]

8.897.2.10 uint8\_t unpack\_nas\_GetServingNetwork\_t::RadiolfacesSize

8.897.2.11 uint32\_t unpack\_nas\_GetServingNetwork\_t::RAN

8.897.2.12 uint32\_t unpack\_nas\_GetServingNetwork\_t::RegistrationState

8.897.2.13 uint32\_t unpack\_nas\_GetServingNetwork\_t::Roaming

## 8.898 unpack\_nas\_GetServingNetworkCapabilities\_t Struct Reference

#### Data Fields

- uint8\_t [DataCapsLen](#)
- uint8\_t [DataCaps](#) [255]

### 8.898.1 Detailed Description

#### Parameters

<i>DataCapsLen</i>	data capabilities len
<i>DataCap</i>	data capabilities

### 8.898.2 Field Documentation

8.898.2.1 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]`

8.898.2.2 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen`

## 8.899 unpack\_nas\_GetSignalStrengths\_t Struct Reference

#### Data Fields

- `uint32_t len`
- signed char `rssi` [8]
- `uint32_t radio` [8]

### 8.899.1 Detailed Description

#### Parameters

<i>len</i>	number of rssi & radio items following
<i>rssi</i>	signal strength array
<i>radio</i>	radio interface array

### 8.899.2 Field Documentation

8.899.2.1 `uint32_t unpack_nas_GetSignalStrengths_t::len`

8.899.2.2 `uint32_t unpack_nas_GetSignalStrengths_t::radio[8]`

8.899.2.3 signed char `unpack_nas_GetSignalStrengths_t::rssi[8]`

## 8.900 unpack\_nas\_PerformNetworkScan\_t Struct Reference

#### Data Fields

- `uint8_t * p3GppNetworkInstanceSize`
- `nas_QmiNas3GppNetworkInfo * p3GppNetworkInfoInstances`
- `uint8_t * pRATInstanceSize`
- `nas_QmiNas3GppNetworkRAT * pRATInstance`
- `uint8_t * pPcsInstanceSize`
- `nas_QmisNasPcsDigit * pPcsInstance`
- `uint32_t * pScanResult`

### 8.900.1 Detailed Description

## Parameters

<i>InstanceSize</i>	total instances
<i>Instances</i>	info for instances

## 8.900.2 Field Documentation

8.900.2.1 nas\_QmiNas3GppNetworkInfo\* unpack\_nas\_PerformNetworkScan\_t::p3GppNetworkInfoInstances

8.900.2.2 uint8\_t\* unpack\_nas\_PerformNetworkScan\_t::p3GppNetworkInstanceSize

8.900.2.3 nas\_QmisNasPcsDigit\* unpack\_nas\_PerformNetworkScan\_t::pPCSInstance

8.900.2.4 uint8\_t\* unpack\_nas\_PerformNetworkScan\_t::pPCSInstanceSize

8.900.2.5 nas\_QmiNas3GppNetworkRAT\* unpack\_nas\_PerformNetworkScan\_t::pRATInstance

8.900.2.6 uint8\_t\* unpack\_nas\_PerformNetworkScan\_t::pRATInstanceSize

8.900.2.7 uint32\_t\* unpack\_nas\_PerformNetworkScan\_t::pScanResult

## 8.901 unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t Struct Reference

## Data Fields

- uint8\_t [dataCapsSize](#)
- uint8\_t [dataCaps](#) [255]

## 8.901.1 Detailed Description

## Parameters

<i>dataCapsSize</i>	Number of Data Capabilities
<i>dataCaps</i>	Data Capabilities

## 8.901.2 Field Documentation

8.901.2.1 uint8\_t unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t::dataCaps[255]

8.901.2.2 uint8\_t unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t::dataCapsSize

## 8.902 unpack\_nas\_SetEventReportInd\_t Struct Reference

## Data Fields

- [nas\\_SignalStrengthTlv](#) SSTlv
- [nas\\_RFInfoTlv](#) RFTlv
- [nas\\_RejectReasonTlv](#) RRTlv
- [nas\\_SLQSSignalStrengthsTlv](#) SLQSSSTlv

## 8.902.1 Detailed Description

## Parameters

<i>SSTlv</i>	signal strength tlv
<i>RFTlv</i>	RF tlv
<i>RRTlv</i>	RR tlv
<i>SLQSSSTlv</i>	signal strength complete info tlv

## 8.902.2 Field Documentation

8.902.2.1 nas\_RFInfoTlv unpack\_nas\_SetEventReportInd\_t::RFTlv

8.902.2.2 nas\_RejectReasonTlv unpack\_nas\_SetEventReportInd\_t::RRTlv

8.902.2.3 nas\_SLQSSignalStrengthsTlv unpack\_nas\_SetEventReportInd\_t::SLQSSSTlv

8.902.2.4 nas\_SignalStrengthTlv unpack\_nas\_SetEventReportInd\_t::SSTlv

## 8.903 unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t Struct Reference

## Data Fields

- [nas\\_PhyCaAggScellIndType](#) sPhyCaAggScellIndType
- [nas\\_PhyCaAggScellIDIBw](#) sPhyCaAggScellIDIBw
- [nas\\_PhyCaAggScellInfo](#) sPhyCaAggScellInfo
- [nas\\_PhyCaAggPcellInfo](#) sPhyCaAggPcellInfo
- [nas\\_PhyCaAggScellIndex](#) sPhyCaAggScellIndex

## 8.903.1 Detailed Description

Structure for storing the LTEC PHY CA indication parameters.

## Parameters

<i>pPhyCaAgg-ScellIndType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndType</a> for more information.</li> </ul>
<i>sPhyCaAgg-ScellIDIBw</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIDIBw</a> for more information.</li> </ul>
<i>sPhyCaAgg-ScellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellInfo</a> for more information.</li> </ul>
<i>sPhyCaAgg-PcellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggPcellInfo</a> for more information.</li> </ul>
<i>sPhyCaAgg-ScellIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndex</a> for more information.</li> </ul>

## 8.903.2 Field Documentation

8.903.2.1 nas\_PhyCaAggPcellInfo unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggPcellInfo

8.903.2.2 nas\_PhyCaAggScellIDBw unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggScellIDBw

8.903.2.3 nas\_PhyCaAggScellIndex unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggScellIndex

8.903.2.4 nas\_PhyCaAggScellIndType unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggScellIndType

8.903.2.5 nas\_PhyCaAggScellInfo unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t::sPhyCaAggScellInfo

## 8.904 unpack\_nas\_SetNetworkPreference\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

### 8.904.1 Detailed Description

#### Parameters

<i>TechnologyPref[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>Duration[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>Persistent-TechnologyPref[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>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.904.2 Field Documentation

8.904.2.1 uint16\_t unpack\_nas\_SetNetworkPreference\_t::Tlvresult

## 8.905 unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [roaming](#)

### 8.905.1 Detailed Description

#### 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>
----------------	--

### 8.905.2 Field Documentation

8.905.2.1 uint8\_t unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t::roaming

## 8.906 unpack\_nas\_SetServingSystemCallback\_ind\_t Struct Reference

### Data Fields

- [NAServingSystemInfo](#) SSInfo
- uint16\_t [Tlvresult](#)

### 8.906.1 Detailed Description

#### Parameters

<i>SSInfo</i>	<ul style="list-style-type: none"><li>• Serving system parameters information<ul style="list-style-type: none"><li>– See <a href="#">NAServingSystemInfo</a> for more details</li></ul></li></ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• unpack result</li></ul>

### 8.906.2 Field Documentation

8.906.2.1 [NAServingSystemInfo](#) unpack\_nas\_SetServingSystemCallback\_ind\_t::SSInfo

8.906.2.2 uint16\_t unpack\_nas\_SetServingSystemCallback\_ind\_t::Tlvresult

## 8.907 unpack\_nas\_SlqsGetLTECphyCAInfo\_t Struct Reference

### Data Fields

- [NasGetLTECphyCAInfo](#) LTECphyCAInfo

- uint16\_t [Tlvresult](#)

### 8.907.1 Detailed Description

#### Parameters

<i>LTECphyCa</i>	<ul style="list-style-type: none"> <li>• Carrier aggregation event information             <ul style="list-style-type: none"> <li>– See <a href="#">NasGetLTECphyCAInfo</a> for more details</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.907.2 Field Documentation

8.907.2.1 [NasGetLTECphyCAInfo](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::LTECphyCAInfo](#)

8.907.2.2 [uint16\\_t](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::Tlvresult](#)

## 8.908 [unpack\\_nas\\_SLQSGetNetworkTime\\_t](#) Struct Reference

#### Data Fields

- [nas\\_timeInfo](#) \* [p3GPP2TimeInfo](#)
- [nas\\_timeInfo](#) \* [p3GPPTimeInfo](#)

### 8.908.1 Detailed Description

This structure contains information about the GetNetworkTime response parameters.

#### Parameters

<i>p3GPP2Time-Info</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_timeInfo</a> for more information</li> </ul>
<i>p3GPPTimeInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_timeInfo</a> for more information</li> </ul>

### 8.908.2 Field Documentation

8.908.2.1 [nas\\_timeInfo\\*](#) [unpack\\_nas\\_SLQSGetNetworkTime\\_t::p3GPP2TimeInfo](#)

8.908.2.2 [nas\\_timeInfo\\*](#) [unpack\\_nas\\_SLQSGetNetworkTime\\_t::p3GPPTimeInfo](#)

## 8.909 [unpack\\_nas\\_SLQSGetPLMNName\\_t](#) Struct Reference

#### Data Fields

- [uint8\\_t](#) [spnEncoding](#)
- [uint8\\_t](#) [spnLength](#)

- char [spn](#) [255]
- uint8\_t [shortNameEn](#)
- uint8\_t [shortNameCI](#)
- uint8\_t [shortNameSB](#)
- char [shortNameLen](#)
- uint8\_t [shortName](#) [255]
- uint8\_t [longNameEn](#)
- uint8\_t [longNameCI](#)
- uint8\_t [longNameSB](#)
- uint8\_t [longNameLen](#)
- char [longName](#) [255]

### 8.909.1 Field Documentation

- 8.909.1.1 char [unpack\\_nas\\_SLQSGetPLMNName\\_t::longName](#)[255]
- 8.909.1.2 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::longNameCI](#)
- 8.909.1.3 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::longNameEn](#)
- 8.909.1.4 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::longNameLen](#)
- 8.909.1.5 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::longNameSB](#)
- 8.909.1.6 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::shortName](#)[255]
- 8.909.1.7 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::shortNameCI](#)
- 8.909.1.8 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::shortNameEn](#)
- 8.909.1.9 char [unpack\\_nas\\_SLQSGetPLMNName\\_t::shortNameLen](#)
- 8.909.1.10 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::shortNameSB](#)
- 8.909.1.11 char [unpack\\_nas\\_SLQSGetPLMNName\\_t::spn](#)[255]
- 8.909.1.12 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::spnEncoding](#)
- 8.909.1.13 uint8\_t [unpack\\_nas\\_SLQSGetPLMNName\\_t::spnLength](#)

## 8.910 unpack\_nas\_SLQSGetservingSystem\_t Struct Reference

### Data Fields

- [nas\\_servSystem](#) [ServingSystem](#)
- uint8\_t [RoamIndicatorVal](#)
- [nas\\_dataSrvCapabilities](#) [DataSrvCapabilities](#)
- [nas\\_currentPLMN](#) [CurrentPLMN](#)
- uint16\_t [SystemID](#)
- uint16\_t [NetworkID](#)
- uint16\_t [BasestationID](#)
- uint32\_t [BasestationLatitude](#)
- uint32\_t [BasestationLongitude](#)
- [nas\\_roamIndList](#) [RoamingIndicatorList](#)

- uint8\_t [DefaultRoamInd](#)
- nas\_qaQmi3Gpp2TimeZone [Gpp2TimeZone](#)
- uint8\_t [CDMA\\_P\\_Rev](#)
- uint8\_t [GppTimeZone](#)
- uint8\_t [GppNetworkDSTAdjustment](#)
- uint16\_t [Lac](#)
- uint32\_t [CellID](#)
- uint8\_t [ConcSvcInfo](#)
- uint8\_t [PRLInd](#)
- uint8\_t [DTMInd](#)
- nas\_detailSvcInfo [DetailedSvcInfo](#)
- nas\_CDMA SysInfoExt [CDMA SystemInfoExt](#)
- uint8\_t [HdrPersonality](#)
- uint16\_t [TrackAreaCode](#)
- nas\_callBarStatus [CallBarStatus](#)

### 8.910.1 Detailed Description

#### Parameters

<i>ServingSystem</i>	serving system info
<i>RoamIndicator-Val</i>	roaming indicator value
<i>DataSrv-Capabilities</i>	data servcie capabilities
<i>CurrentPLMN</i>	current PLMN info
<i>SystemID</i>	system id
<i>NetworkID</i>	network id
<i>BasestationID</i>	base station id
<i>Basestation-Latitude</i>	base station latitude
<i>Basestation-Longitude</i>	base station longitude
<i>Roaming-IndicatorList</i>	roaming indicator list
<i>DefaultRoamInd</i>	default roaming indicator
<i>3Gpp2TimeZone</i>	3Gpp2 time zone
<i>pCDMA_P_Rev</i>	cdma P_Rev in use
<i>3GppTimeZone</i>	3Gpp time zone
<i>GppNetworkDSTAdjustment</i>	3GPP network daylight saving adjustment
<i>Lac</i>	location area code
<i>CellID</i>	3GPP cell id
<i>ConcSvcInfo</i>	3GPP2 concurrent servcie info
<i>PRLInd</i>	3GPP2 PRL indicator
<i>DTMInd</i>	DTM indicator(GSM)
<i>DetailedSvcInfo</i>	detail servcie info
<i>CDMA System-InfoExt</i>	extra cdma system info
<i>HdrPersonality</i>	hdr personality
<i>TrackAreaCode</i>	track area code
<i>CallBarStatus</i>	call barring status

### 8.910.2 Field Documentation

- 8.910.2.1 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::BasestationID
- 8.910.2.2 uint32\_t unpack\_nas\_SLQSGetServingSystem\_t::BasestationLatitude
- 8.910.2.3 uint32\_t unpack\_nas\_SLQSGetServingSystem\_t::BasestationLongitude
- 8.910.2.4 nas\_callBarStatus unpack\_nas\_SLQSGetServingSystem\_t::CallBarStatus
- 8.910.2.5 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::CDMA\_P\_Rev
- 8.910.2.6 nas\_CDMASysInfoExt unpack\_nas\_SLQSGetServingSystem\_t::CDMASystemInfoExt
- 8.910.2.7 uint32\_t unpack\_nas\_SLQSGetServingSystem\_t::CellID
- 8.910.2.8 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::ConcSvcInfo
- 8.910.2.9 nas\_currentPLMN unpack\_nas\_SLQSGetServingSystem\_t::CurrentPLMN
- 8.910.2.10 nas\_dataSrvCapabilities unpack\_nas\_SLQSGetServingSystem\_t::DataSrvCapabilities
- 8.910.2.11 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::DefaultRoamInd
- 8.910.2.12 nas\_detailSvcInfo unpack\_nas\_SLQSGetServingSystem\_t::DetailedSvcInfo
- 8.910.2.13 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::DTMInd
- 8.910.2.14 nas\_qaQmi3Gpp2TimeZone unpack\_nas\_SLQSGetServingSystem\_t::Gpp2TimeZone
- 8.910.2.15 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::GppNetworkDSTAdjustment
- 8.910.2.16 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::GppTimeZone
- 8.910.2.17 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::HdrPersonality
- 8.910.2.18 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::Lac
- 8.910.2.19 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::NetworkID
- 8.910.2.20 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::PRLInd
- 8.910.2.21 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::RoamIndicatorVal
- 8.910.2.22 nas\_roamIndList unpack\_nas\_SLQSGetServingSystem\_t::RoamingIndicatorList
- 8.910.2.23 nas\_servSystem unpack\_nas\_SLQSGetServingSystem\_t::ServingSystem
- 8.910.2.24 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::SystemID
- 8.910.2.25 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::TrackAreaCode

## 8.911 unpack\_nas\_SLQSGetSignalStrength\_t Struct Reference

### Data Fields

- uint16\_t [signalStrengthReqMask](#)
- uint16\_t [rxSignalStrengthListLen](#)

- [nas\\_rxSignalStrengthListElement](#) rxSignalStrengthList [18]
- [uint16\\_t](#) ecioListLen
- [nas\\_ecioListElement](#) ecioList [18]
- [int32\\_t](#) lo
- [uint8\\_t](#) sinr
- [uint16\\_t](#) errorRateListLen
- [nas\\_errorRateListElement](#) errorRateList [18]
- [nas\\_rsrqInformation](#) rsrqInfo
- [int16\\_t](#) ltesnr
- [int16\\_t](#) ltersrp

### 8.911.1 Detailed Description

#### Parameters

<i>rxSignalStrengthListLen</i>	number of elements in Receive Signal Strength List
<i>rxSignalStrengthList</i>	signal strength list
<i>ecioListLen</i>	number of elements in ECIO List
<i>ecioList</i>	ecio list
<i>lo</i>	received lo in dBm; IO is only applicable for 1xEV-DO
<i>sinr</i>	SINR level; SINR is only applicable for 1xEV-DO
<i>errorRateListLen</i>	number of elements in Error Rate List
<i>errorRateList</i>	error rate list
<i>rsrqInfo</i>	rsrq info
<i>ltesnr</i>	lte snr info
<i>ltersrp</i>	lte srp info

### 8.911.2 Field Documentation

8.911.2.1 [nas\\_ecioListElement](#) unpack\_nas\_SLQSGetSignalStrength\_t::ecioList[18]

8.911.2.2 [uint16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::ecioListLen

8.911.2.3 [nas\\_errorRateListElement](#) unpack\_nas\_SLQSGetSignalStrength\_t::errorRateList[18]

8.911.2.4 [uint16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::errorRateListLen

8.911.2.5 [int32\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::lo

8.911.2.6 [int16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::ltersrp

8.911.2.7 [int16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::ltesnr

8.911.2.8 [nas\\_rsrqInformation](#) unpack\_nas\_SLQSGetSignalStrength\_t::rsrqInfo

8.911.2.9 [nas\\_rxSignalStrengthListElement](#) unpack\_nas\_SLQSGetSignalStrength\_t::rxSignalStrengthList[18]

8.911.2.10 [uint16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::rxSignalStrengthListLen

8.911.2.11 [uint16\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::signalStrengthReqMask

8.911.2.12 [uint8\\_t](#) unpack\_nas\_SLQSGetSignalStrength\_t::sinr

## 8.912 unpack\_nas\_SLQSGetSysInfo\_t Struct Reference

### Data Fields

- [nas\\_SrvStatusInfo](#) \* [pCDMASrvStatusInfo](#)
- [nas\\_SrvStatusInfo](#) \* [pHDRSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pGSMSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pWCDMASrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pLTESrvStatusInfo](#)
- [nas\\_CDMASysInfo](#) \* [pCDMASysInfo](#)
- [nas\\_HDRSysInfo](#) \* [pHDRSysInfo](#)
- [nas\\_GSMSysInfo](#) \* [pGSMSysInfo](#)
- [nas\\_WCDMASysInfo](#) \* [pWCDMASysInfo](#)
- [nas\\_LTESysInfo](#) \* [pLTESysInfo](#)
- [nas\\_AddCDMASysInfo](#) \* [pAddCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddHDRSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddGSMSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddLTESysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [uint8\\_t](#) \* [pLTEVoiceSupportSysInfo](#)
- [uint8\\_t](#) \* [pGSMCipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pWCDMACipherDomainSysInfo](#)

### 8.912.1 Detailed Description

#### 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>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>pAddGSM Sys- 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>

### 8.912.2 Field Documentation

- 8.912.2.1 nas\_AddCDMASysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pAddCDMASysInfo
- 8.912.2.2 nas\_AddSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pAddGSM SysInfo
- 8.912.2.3 uint16\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pAddHDR SysInfo
- 8.912.2.4 uint16\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pAddLTE SysInfo
- 8.912.2.5 nas\_AddSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pAddWCDMA SysInfo
- 8.912.2.6 nas\_SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pCDMA SrvStatusInfo
- 8.912.2.7 nas\_CDMA SysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pCDMA SysInfo
- 8.912.2.8 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSM CallBarringSysInfo
- 8.912.2.9 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pGSM CipherDomainSysInfo
- 8.912.2.10 nas\_GSM SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSM SrvStatusInfo
- 8.912.2.11 nas\_GSM SysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSM SysInfo
- 8.912.2.12 nas\_SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDR SrvStatusInfo
- 8.912.2.13 nas\_HDR SysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDR SysInfo
- 8.912.2.14 nas\_GSM SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTE SrvStatusInfo
- 8.912.2.15 nas\_LTE SysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTE SysInfo
- 8.912.2.16 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pLTE VoiceSupportSysInfo
- 8.912.2.17 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMA CallBarringSysInfo
- 8.912.2.18 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMA CipherDomainSysInfo
- 8.912.2.19 nas\_GSM SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMA SrvStatusInfo
- 8.912.2.20 nas\_WCDMA SysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMA SysInfo

## 8.913 unpack\_nas\_SLQSGetSysSelectionPref\_t Struct Reference

### Data Fields

- uint8\_t \* [pEmerMode](#)
- uint16\_t \* [pModePref](#)
- uint64\_t \* [pBandPref](#)
- uint16\_t \* [pPRLPref](#)
- uint16\_t \* [pRoamPref](#)
- uint64\_t \* [pLTEBandPref](#)
- uint8\_t \* [pNetSelPref](#)
- uint32\_t \* [pSrvDomainPref](#)
- uint32\_t \* [pGWAcqOrderPref](#)

### 8.913.1 Detailed Description

#### 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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</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> <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> <li>– All other bits are reserved</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> </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>

## 8.913.2 Field Documentation

8.913.2.1 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pBandPref

8.913.2.2 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pEmerMode

8.913.2.3 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pGWAcqOrderPref

8.913.2.4 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pLTEBandPref

8.913.2.5 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pModePref

8.913.2.6 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pNetSelPref

8.913.2.7 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pPRLPref

8.913.2.8 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pRoamPref

8.913.2.9 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pSrvDomainPref

## 8.914 unpack\_nas\_SLQSNasGetCellLocationInfo\_t Struct Reference

### Data Fields

- [nas\\_GERANInfo](#) \* [pGERANInfo](#)
- [nas\\_UMTSInfo](#) \* [pUMTSInfo](#)
- [nas\\_CDMAInfo](#) \* [pCDMAInfo](#)
- [nas\\_LTEInfoIntrafreq](#) \* [pLTEInfoIntrafreq](#)
- [nas\\_LTEInfoInterfreq](#) \* [pLTEInfoInterfreq](#)

- [nas\\_LTEInfoNeighboringGSM](#) \* [pLTEInfoNeighboringGSM](#)
- [nas\\_LTEInfoNeighboringWCDMA](#) \* [pLTEInfoNeighboringWCDMA](#)
- [uint32\\_t](#) \* [pUMTSCellID](#)
- [nas\\_WCDMAInfoLTENeighborCell](#) \* [pWCDMAInfoLTENeighborCell](#)

### 8.914.1 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="#">nas_GERANInfo</a> for more information.</li> </ul>
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UMTSInfo</a> for more information.</li> </ul>
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMAInfo</a> for more information.</li> </ul>
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoIntrafreq</a> for more information.</li> </ul>
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoInterfreq</a> for more information.</li> </ul>
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringGSM</a> for more information.</li> </ul>
<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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>pWCDMAInfoLTENeighborCell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMAInfoLTENeighborCell</a> for more information.</li> </ul>

### 8.914.2 Field Documentation

8.914.2.1 [nas\\_CDMAInfo](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pCDMAInfo](#)

8.914.2.2 [nas\\_GERANInfo](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pGERANInfo](#)

8.914.2.3 [nas\\_LTEInfoInterfreq](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pLTEInfoInterfreq](#)

8.914.2.4 [nas\\_LTEInfoIntrafreq](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pLTEInfoIntrafreq](#)

8.914.2.5 [nas\\_LTEInfoNeighboringGSM](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pLTEInfoNeighboringGSM](#)

8.914.2.6 [nas\\_LTEInfoNeighboringWCDMA](#)\* [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t::pLTEInfoNeighboringWCDMA](#)

8.914.2.7 uint32\_t\* unpack\_nas\_SLQSNasGetCellLocationInfo\_t::pUMTSCellID

8.914.2.8 nas\_UMTSInfo\* unpack\_nas\_SLQSNasGetCellLocationInfo\_t::pUMTSInfo

8.914.2.9 nas\_WCDMAInfoLTENeighborCell\* unpack\_nas\_SLQSNasGetCellLocationInfo\_t::pWCDMAInfoLTENeighborCell

## 8.915 unpack\_nas\_SLQSNasGetSigInfo\_t Struct Reference

### Data Fields

- [cdmaSSInfo](#) [CDMASSInfo](#)
- [hdrSSInfo](#) [HDRSSInfo](#)
- [int8\\_t](#) [GSMSSInfo](#)
- [cdmaSSInfo](#) [WCDMASSInfo](#)
- [lteSSInfo](#) [LTESSInfo](#)

### 8.915.1 Detailed Description

#### Parameters

<a href="#">CDMASSInfo</a>	CDMA Signal Strength Information
<a href="#">HDRSSInfo</a>	HDR Signal Strength Information
<a href="#">GSMSSInfo</a>	GSM signal strength is the RSSI in dBm.
<a href="#">WCDMASSInfo</a>	WCDMA Signal Strength Information
<a href="#">LTESSInfo</a>	LTE Signal Strength Information

### 8.915.2 Field Documentation

8.915.2.1 [cdmaSSInfo](#) [unpack\\_nas\\_SLQSNasGetSigInfo\\_t::CDMASSInfo](#)

8.915.2.2 [int8\\_t](#) [unpack\\_nas\\_SLQSNasGetSigInfo\\_t::GSMSSInfo](#)

8.915.2.3 [hdrSSInfo](#) [unpack\\_nas\\_SLQSNasGetSigInfo\\_t::HDRSSInfo](#)

8.915.2.4 [lteSSInfo](#) [unpack\\_nas\\_SLQSNasGetSigInfo\\_t::LTESSInfo](#)

8.915.2.5 [cdmaSSInfo](#) [unpack\\_nas\\_SLQSNasGetSigInfo\\_t::WCDMASSInfo](#)

## 8.916 unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t Struct Reference

### Data Fields

- [nas\\_UniversalTime](#) [universalTime](#)
- [uint8\\_t](#) \* [pTimeZone](#)
- [uint8\\_t](#) \* [pDayltSavAdj](#)
- [uint8\\_t](#) \* [pRadioInterface](#)

### 8.916.1 Detailed Description

Structure for storing the NAS Network Time indication parameters.

## Parameters

<i>universalTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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>
<i>pRadioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface from which the information comes</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x01 - NAS_RADIO_IF_CDMA_1X - cdma2000 1X</li> <li>– 0x02 - NAS_RADIO_IF_CDMA_1xEVDO - 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>

## 8.916.2 Field Documentation

8.916.2.1 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pDayltSavAdj

8.916.2.2 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pRadioInterface

8.916.2.3 uint8\_t\* unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::pTimeZone

8.916.2.4 nas\_UniversalTime unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t::universalTime

## 8.917 unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t Struct Reference

## Data Fields

- [cdmaSSInfo](#) \* [pCDMASigInfo](#)
- [hdrSSInfo](#) \* [pHDRSigInfo](#)
- [int8\\_t](#) \* [pGSMSigInfo](#)
- [cdmaSSInfo](#) \* [pWCDMASigInfo](#)
- [lteSSInfo](#) \* [pLTESigInfo](#)
- [int8\\_t](#) \* [pRscp](#)
- [tdscdmaSigInfoExt](#) \* [pTDSCDMASigInfoExt](#)

## 8.917.1 Detailed Description

## Parameters

<i>pCDMASigInfo</i>	CDMA SS info
<i>pHDRSigInfo</i>	HDR SS info
<i>pGSMSigInfo</i>	GSM signal info
<i>pWCDMASigInfo</i>	WCDMA signal info
<i>pLTESigInfo</i>	LTE signal info
<i>pRscp</i>	RSCP of the Primary Common Control Physical Channel
<i>pTDSCDMASigInfoExt</i>	extra CDMA sig info

## 8.917.2 Field Documentation

8.917.2.1 **cdmaSSInfo\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pCDMASigInfo8.917.2.2 **int8\_t\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pGSMSigInfo8.917.2.3 **hdrSSInfo\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pHDRSigInfo8.917.2.4 **lteSSInfo\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pLTESigInfo8.917.2.5 **int8\_t\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pRscp8.917.2.6 **tdscdmaSigInfoExt\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pTDSCDMASigInfoExt8.917.2.7 **cdmaSSInfo\*** unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t::pWCDMASigInfo

## 8.918 unpack\_nas\_SLQSNasSwiModemStatus\_t Struct Reference

## Data Fields

- [nas\\_CommlInfo](#) **commonInfo**
- [nas\\_LTEInfo](#) \* **pLTEInfo**

## 8.918.1 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>

## 8.918.2 Field Documentation

8.918.2.1 **nas\_CommlInfo** unpack\_nas\_SLQSNasSwiModemStatus\_t::commonInfo8.918.2.2 **nas\_LTEInfo\*** unpack\_nas\_SLQSNasSwiModemStatus\_t::pLTEInfo

## 8.919 unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t Struct Reference

### Data Fields

- [NASQmiCbkNasSwiOTAMessageInd](#) Info
- uint16\_t [Tlvresult](#)

### 8.919.1 Detailed Description

#### Parameters

<i>Info</i>	<ul style="list-style-type: none"> <li>• Structure used to store all QMI Notification Info. <ul style="list-style-type: none"> <li>– See <a href="#">NASQmiCbkNasSwiOTAMessageInd</a> for more details</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.919.2 Field Documentation

8.919.2.1 [NASQmiCbkNasSwiOTAMessageInd](#) `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Info`

8.919.2.2 `uint16_t` `unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t::Tlvresult`

## 8.920 unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t Struct Reference

### Data Fields

- [NASQmiCbkNasSystemSelPrefInd](#) Info
- uint16\_t [Tlvresult](#)

### 8.920.1 Detailed Description

#### Parameters

<i>Info</i>	<ul style="list-style-type: none"> <li>• Structure used to store all QMI Notification Info. <ul style="list-style-type: none"> <li>– See <a href="#">NASQmiCbkNasSystemSelPrefInd</a> for more details</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.920.2 Field Documentation

8.920.2.1 [NASQmiCbkNasSystemSelPrefInd](#) `unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Info`

8.920.2.2 `uint16_t` `unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t::Tlvresult`

## 8.921 unpack\_nas\_SLQSSwiGetLteCQI\_t Struct Reference

## Data Fields

- uint8\_t [ValidityCW0](#)
- uint8\_t [CQIValueCW0](#)
- uint8\_t [ValidityCW1](#)
- uint8\_t [CQIValueCW1](#)

## 8.921.1 Detailed Description

## Parameters

<i>ValidityCW0[OUT]</i>	<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>CQIValueCW0[OUT]</i>	<ul style="list-style-type: none"> <li>• Values             <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>
<i>ValidityCW1[OUT]</i>	<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>CQIValueCW1[OUT]</i>	<ul style="list-style-type: none"> <li>• Values             <ul style="list-style-type: none"> <li>– Range 0~15</li> </ul> </li> </ul>

## 8.921.2 Field Documentation

8.921.2.1 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::CQIValueCW0

8.921.2.2 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::CQIValueCW1

8.921.2.3 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::ValidityCW0

8.921.2.4 uint8\_t unpack\_nas\_SLQSSwiGetLteCQI\_t::ValidityCW1

## 8.922 unpack\_nas\_SLQSSysInfoCallback\_ind\_t Struct Reference

## Data Fields

- [nas\\_SrvStatusInfo](#) \* [pCDMASrvStatusInfo](#)
- [nas\\_SrvStatusInfo](#) \* [pHDRSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pGSMSrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pWCDMASrvStatusInfo](#)
- [nas\\_GSMSrvStatusInfo](#) \* [pLTESrvStatusInfo](#)
- [nas\\_CDMASysInfo](#) \* [pCDMASysInfo](#)
- [nas\\_HDRSysInfo](#) \* [pHDRSysInfo](#)
- [nas\\_GSMSysInfo](#) \* [pGSMSysInfo](#)
- [nas\\_WCDMASysInfo](#) \* [pWCDMASysInfo](#)

- [nas\\_LTESysInfo](#) \* [pLTESysInfo](#)
- [nas\\_AddCDMASysInfo](#) \* [pAddCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddHDRSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddGSM SysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddLTESysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [uint8\\_t](#) \* [pLTVoiceSupportSysInfo](#)
- [uint8\\_t](#) \* [pGSMCipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pWCDMACipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pSysInfoNoChange](#)

### 8.922.1 Detailed Description

#### 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>pGSMCallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">CallBarringSysInfo</a> for more information.</li> </ul>
<i>pWCDMACallBarringSysInfo</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>pGSMCipherDomainSysInfo</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>

## 8.922.2 Field Documentation

8.922.2.1 **nas\_AddCDMASysInfo\*** unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddCDMASysInfo

8.922.2.2 **nas\_AddSysInfo\*** unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddGSMSysInfo

8.922.2.3 **uint16\_t\*** unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddHDRSysInfo

8.922.2.4 **uint16\_t\*** unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddLTESysInfo

- 8.922.2.5 `nas_AddSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pAddWCDMASysInfo`
- 8.922.2.6 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASrvStatusInfo`
- 8.922.2.7 `nas_CDMASysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pCDMASysInfo`
- 8.922.2.8 `nas_CallBarringSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCallBarringSysInfo`
- 8.922.2.9 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMCipherDomainSysInfo`
- 8.922.2.10 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSrvStatusInfo`
- 8.922.2.11 `nas_GSMSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pGSMSysInfo`
- 8.922.2.12 `nas_SrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDSrvStatusInfo`
- 8.922.2.13 `nas_HDRSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pHDSysInfo`
- 8.922.2.14 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTESrvStatusInfo`
- 8.922.2.15 `nas_LTESysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTESysInfo`
- 8.922.2.16 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pLTEVoiceSupportSysInfo`
- 8.922.2.17 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pSysInfoNoChange`
- 8.922.2.18 `nas_CallBarringSysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACallBarringSysInfo`
- 8.922.2.19 `uint8_t*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMACipherDomainSysInfo`
- 8.922.2.20 `nas_GSMSrvStatusInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASrvStatusInfo`
- 8.922.2.21 `nas_WCDMASysInfo*` `unpack_nas_SLQSSysInfoCallback_ind_t::pWCDMASysInfo`

## 8.923 `unpack_omaDmConfigTlv_t` Struct Reference

### Data Fields

- `uint8_t` [state](#)
- `uint8_t` [userInputReq](#)
- `uint16_t` [userInputTimeout](#)
- `uint16_t` [alertmsglength](#)
- `uint8_t` [alertmsg](#) [256]

### 8.923.1 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>	<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>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>

### 8.923.2 Field Documentation

8.923.2.1 `uint8_t unpack_omaDmConfigTlv_t::alertmsg[256]`

8.923.2.2 `uint16_t unpack_omaDmConfigTlv_t::alertmsglength`

8.923.2.3 `uint8_t unpack_omaDmConfigTlv_t::state`

8.923.2.4 `uint8_t unpack_omaDmConfigTlv_t::userInputReq`

8.923.2.5 `uint16_t unpack_omaDmConfigTlv_t::userInputTimeout`

## 8.924 unpack\_omaDmFotaTlv\_t Struct Reference

### Data Fields

- `uint8_t state`
- `uint8_t userInputReq`
- `uint16_t userInputTimeout`
- `uint32_t fwdloadsize`
- `uint32_t fwloadComplete`
- `uint16_t updateCompleteStatus`
- `uint8_t severity`
- `uint16_t versionlength`
- `uint8_t version [256]`
- `uint16_t namelength`
- `uint8_t package_name [256]`
- `uint16_t descriptionlength`
- `uint8_t description [256]`
- `uint8_t sessionType`

### 8.924.1 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>

### 8.924.2 Field Documentation

- 8.924.2.1 `uint8_t` `unpack_omaDmFotaTlv_t::description[256]`
- 8.924.2.2 `uint16_t` `unpack_omaDmFotaTlv_t::descriptionlength`
- 8.924.2.3 `uint32_t` `unpack_omaDmFotaTlv_t::fwdloadsize`
- 8.924.2.4 `uint32_t` `unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.924.2.5 `uint16_t` `unpack_omaDmFotaTlv_t::namelength`
- 8.924.2.6 `uint8_t` `unpack_omaDmFotaTlv_t::package_name[256]`
- 8.924.2.7 `uint8_t` `unpack_omaDmFotaTlv_t::sessionType`
- 8.924.2.8 `uint8_t` `unpack_omaDmFotaTlv_t::severity`
- 8.924.2.9 `uint8_t` `unpack_omaDmFotaTlv_t::state`
- 8.924.2.10 `uint16_t` `unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.924.2.11 `uint8_t` `unpack_omaDmFotaTlv_t::userInputReq`
- 8.924.2.12 `uint16_t` `unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.924.2.13 `uint8_t` `unpack_omaDmFotaTlv_t::version[256]`
- 8.924.2.14 `uint16_t` `unpack_omaDmFotaTlv_t::versionlength`

## 8.925 unpack\_omaDmNotificationsTlv\_t Struct Reference

### Data Fields

- `uint8_t` [notification](#)
- `uint16_t` [sessionStatus](#)

### 8.925.1 Field Documentation

- 8.925.1.1 `uint8_t` `unpack_omaDmNotificationsTlv_t::notification`
- 8.925.1.2 `uint16_t` `unpack_omaDmNotificationsTlv_t::sessionStatus`

## 8.926 unpack\_qmi\_t Struct Reference

### Data Fields

- `enum` [msgtype type](#)
- `uint16_t` [msgid](#)
- `uint16_t` [xid](#)

### 8.926.1 Detailed Description

qmi response context

#### Parameters

out	<i>type</i>	message type
out	<i>msgid</i>	message id
out	<i>xid</i>	transaction id

### 8.926.2 Field Documentation

8.926.2.1 uint16\_t unpack\_qmi\_t::msgid

8.926.2.2 enum msgtype unpack\_qmi\_t::type

8.926.2.3 uint16\_t unpack\_qmi\_t::xid

## 8.927 unpack\_qos\_dataRate\_t Struct Reference

### Data Fields

- uint32\_t [dataRateMax](#)
- uint32\_t [guaranteedRate](#)

### 8.927.1 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)

### 8.927.2 Field Documentation

8.927.2.1 uint32\_t unpack\_qos\_dataRate\_t::dataRateMax

8.927.2.2 uint32\_t unpack\_qos\_dataRate\_t::guaranteedRate

## 8.928 unpack\_qos\_IPv4Addr\_t Struct Reference

### Data Fields

- uint32\_t [addr](#)
- uint32\_t [subnetMask](#)

### 8.928.1 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>

## 8.928.2 Field Documentation

8.928.2.1 uint32\_t unpack\_qos\_IPv4Addr\_t::addr

8.928.2.2 uint32\_t unpack\_qos\_IPv4Addr\_t::subnetMask

## 8.929 unpack\_qos\_IPv6Addr\_t Struct Reference

## Data Fields

- uint8\_t [addr](#) [16]
- uint8\_t [prefixLen](#)

## 8.929.1 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.

## 8.929.2 Field Documentation

8.929.2.1 uint8\_t unpack\_qos\_IPv6Addr\_t::addr[16]

8.929.2.2 uint8\_t unpack\_qos\_IPv6Addr\_t::prefixLen

## 8.930 unpack\_qos\_IPv6TrafCls\_t Struct Reference

## Data Fields

- uint8\_t [val](#)
- uint8\_t [mask](#)

## 8.930.1 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) 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>

## 8.930.2 Field Documentation

8.930.2.1 uint8\_t unpack\_qos\_IPv6TrafCls\_t::mask

8.930.2.2 uint8\_t unpack\_qos\_IPv6TrafCls\_t::val

## 8.931 unpack\_qos\_pktErrRate\_t Struct Reference

## Data Fields

- uint16\_t [multiplier](#)
- uint16\_t [exponent](#)

## 8.931.1 Detailed Description

This structure contains the IP flow packet error rate

## Parameters

<i>multiplier</i>	Factor m in calculating packet error rate: $E = m * 10^{**(-p)}$
<i>exponent</i>	Factor p in calculating packet error rate (see above)

## 8.931.2 Field Documentation

8.931.2.1 uint16\_t unpack\_qos\_pktErrRate\_t::exponent

8.931.2.2 uint16\_t unpack\_qos\_pktErrRate\_t::multiplier

## 8.932 unpack\_qos\_Port\_t Struct Reference

## Data Fields

- uint16\_t [port](#)
- uint16\_t [range](#)

## 8.932.1 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

## 8.932.2 Field Documentation

8.932.2.1 uint16\_t unpack\_qos\_Port\_t::port

8.932.2.2 uint16\_t unpack\_qos\_Port\_t::range

## 8.933 unpack\_qos\_QosFlowInfo\_t Struct Reference

## Data Fields

- [unpack\\_qos\\_QosFlowInfoState\\_t](#) QFlowState
- [uint8\\_t is\\_TxQFlowGranted\\_Available](#)
- [unpack\\_qos\\_swiQosFlow\\_t](#) TxQFlowGranted
- [uint8\\_t is\\_RxQFlowGranted\\_Available](#)
- [unpack\\_qos\\_swiQosFlow\\_t](#) RxQFlowGranted
- [uint8\\_t](#) NumTxFilters
- [unpack\\_qos\\_swiQosFilter\\_t](#) TxQFilter [25]
- [uint8\\_t](#) NumRxFilters
- [unpack\\_qos\\_swiQosFilter\\_t](#) RxQFilter [25]
- [uint8\\_t](#) BearerID

## 8.933.1 Detailed Description

Structure with QoS flow details.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

## Parameters

<i>QFlowState</i>	<ul style="list-style-type: none"> <li>• QoS flow state information, please check <a href="#">unpack_qos_QosFlowInfoState_t</a> for more information</li> </ul>
<i>is_TxQFlow-Granted_Available</i>	<ul style="list-style-type: none"> <li>• TRUE if optional TxQFlowGranted is available</li> </ul>
<i>TxQFlow-Granted</i>	<ul style="list-style-type: none"> <li>• The Tx Qos flow granted, please check <a href="#">unpack_qos_swiQosFlow_t</a> for more information</li> </ul>
<i>is_RxQFlow-Granted_Available</i>	<ul style="list-style-type: none"> <li>• TRUE if optional RxQFlowGranted is available</li> </ul>
<i>RxQFlow-Granted</i>	<ul style="list-style-type: none"> <li>• The Rx Qos flow granted, please check <a href="#">unpack_qos_swiQosFlow_t</a> for more information</li> </ul>
<i>NumTxFilters</i>	<ul style="list-style-type: none"> <li>• Number of Tx filters available</li> </ul>

<i>TxQFilter</i>	<ul style="list-style-type: none"> <li>• The Tx Qos filter, please check <a href="#">unpack_qos_swiQosFilter_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FILTERS</a> for more information</li> </ul>
<i>NumRxFilters</i>	<ul style="list-style-type: none"> <li>• Number of Tx filters available</li> </ul>
<i>RxQFilter</i>	<ul style="list-style-type: none"> <li>• The Rx Qos filter, please check <a href="#">unpack_qos_swiQosFilter_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FILTERS</a> for more information</li> </ul>
<i>BearerID</i>	<ul style="list-style-type: none"> <li>• 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>

### 8.933.2 Field Documentation

- 8.933.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`
- 8.933.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`
- 8.933.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`
- 8.933.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`
- 8.933.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`
- 8.933.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`
- 8.933.2.7 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]`
- 8.933.2.8 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted`
- 8.933.2.9 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]`
- 8.933.2.10 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted`

## 8.934 unpack\_qos\_QosFlowInfoState\_t Struct Reference

### Data Fields

- `uint32_t id`
- `uint8_t isNewFlow`
- `uint8_t state`

### 8.934.1 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>	This indicates that the flow that was added/modified/deleted: <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>

## 8.934.2 Field Documentation

8.934.2.1 uint32\_t unpack\_qos\_QoSFlowInfoState\_t::id

8.934.2.2 uint8\_t unpack\_qos\_QoSFlowInfoState\_t::isNewFlow

8.934.2.3 uint8\_t unpack\_qos\_QoSFlowInfoState\_t::state

## 8.935 unpack\_qos\_SLQSQoSGetNetworkStatus\_t Struct Reference

## Data Fields

- uint8\_t [NWQoSStatus](#)

## 8.935.1 Detailed Description

Structure that contains the response to get NW QoS status command

## Parameters

<i>NWQoSStatus</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>
--------------------	--

## 8.935.2 Field Documentation

8.935.2.1 uint8\_t unpack\_qos\_SLQSQoSGetNetworkStatus\_t::NWQoSStatus

## 8.936 unpack\_qos\_SLQSQoSswiReadApnExtraParams\_t Struct Reference

## Data Fields

- uint32\_t [apnId](#)
- uint8\_t [ambr\\_ul](#)
- uint8\_t [ambr\\_dl](#)
- uint8\_t [ambr\\_ul\\_ext](#)

- [uint8\\_t ambr\\_dl\\_ext](#)
- [uint8\\_t ambr\\_ul\\_ext2](#)
- [uint8\\_t ambr\\_dl\\_ext2](#)

### 8.936.1 Detailed Description

Structure that contains extra APN parameters

#### 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>

### 8.936.2 Field Documentation

8.936.2.1 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl`

8.936.2.2 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext`

8.936.2.3 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_dl_ext2`

8.936.2.4 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul`

8.936.2.5 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext`

8.936.2.6 `uint8_t unpack_qos_SLQSQoSswiReadApnExtraParams_t::ambr_ul_ext2`

8.936.2.7 uint32\_t unpack\_qos\_SLQSQoSswiReadApnExtraParams\_t::apnId

## 8.937 unpack\_qos\_SLQSQoSswiReadDataStats\_t Struct Reference

### Data Fields

- uint32\_t [apnId](#)
- uint32\_t [total\\_tx\\_pkt](#)
- uint32\_t [total\\_tx\\_pkt\\_drp](#)
- uint32\_t [total\\_rx\\_pkt](#)
- uint64\_t [total\\_tx\\_bytes](#)
- uint64\_t [total\\_tx\\_bytes\\_drp](#)
- uint64\_t [total\\_rx\\_bytes](#)
- uint32\_t [numQoSFlow](#)
- [unpack\\_QoSFlowStat\\_t](#) [qosFlow](#) [10]

### 8.937.1 Detailed Description

Structure that contains APN data statistics

#### 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[LIBPACK_MAX_QOS_FLOW_PER_APN_STATS]</i>	<ul style="list-style-type: none"> <li>• Data statistic per QoS flow</li> <li>• See <a href="#">unpack_QoSFlowStat_t</a> for more information</li> <li>• See <a href="#">LIBPACK_MAX_QOS_FLOW_PER_APN_STATS</a> for more information</li> </ul>

### 8.937.2 Field Documentation

- 8.937.2.1 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::apnId`
- 8.937.2.2 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::numQosFlow`
- 8.937.2.3 `unpack_QosFlowStat_t unpack_qos_SLQSQosSwiReadDataStats_t::qosFlow[10]`
- 8.937.2.4 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_bytes`
- 8.937.2.5 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_rx_pkt`
- 8.937.2.6 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes`
- 8.937.2.7 `uint64_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_bytes_drp`
- 8.937.2.8 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt`
- 8.937.2.9 `uint32_t unpack_qos_SLQSQosSwiReadDataStats_t::total_tx_pkt_drp`

## 8.938 `unpack_qos_SLQSSetQosEventCallback_ind_t` Struct Reference

### Data Fields

- `uint8_t NumFlows`
- `unpack_qos_QosFlowInfo_t QosFlowInfo` [8]

### 8.938.1 Detailed Description

Structure with QoS event details

#### Parameters

<i>NumFlows</i>	<ul style="list-style-type: none"> <li>• Number of QoS flows available</li> </ul>
<i>QosFlowInfo</i>	<ul style="list-style-type: none"> <li>• The Qos flow details, please check <code>unpack_qos_QosFlowInfo_t</code> for more information</li> <li>• See <code>LIBPACK_MAX_QOS_FLOWS</code> for more information</li> </ul>

### 8.938.2 Field Documentation

- 8.938.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`
- 8.938.2.2 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

## 8.939 `unpack_qos_SLQSSetQosNWStatusCallback_ind_t` Struct Reference

### Data Fields

- `uint8_t status`

### 8.939.1 Detailed Description

Structure with network's QoS status

#### 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>
---------------	--

#### Note

- Technology Supported: CDMA

### 8.939.2 Field Documentation

8.939.2.1 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

## 8.940 unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t Struct Reference

#### Data Fields

- `uint16_t event`

### 8.940.1 Detailed Description

Structure with QoS primary flow events

#### 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>
--------------	---

### 8.940.2 Field Documentation

8.940.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

## 8.941 unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t Struct Reference

#### Data Fields

- `uint32_t id`
- `uint8_t status`
- `uint8_t event`
- `uint8_t reason`

### 8.941.1 Detailed Description

Structure with QoS status indication details

## Parameters

<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>

## 8.941.2 Field Documentation

8.941.2.1 uint8\_t unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t::event

8.941.2.2 uint32\_t unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t::id

8.941.2.3 uint8\_t unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t::reason

8.941.2.4 uint8\_t unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t::status

## 8.942 unpack\_qos\_swiQosFilter\_t Struct Reference

### Data Fields

- uint8\_t [index](#)
- uint8\_t [version](#)
- uint8\_t [is\\_IPv4SrcAddr\\_Available](#)
- [unpack\\_qos\\_IPv4Addr\\_t](#) [IPv4SrcAddr](#)
- uint8\_t [is\\_IPv4DstAddr\\_Available](#)
- [unpack\\_qos\\_IPv4Addr\\_t](#) [IPv4DstAddr](#)
- uint8\_t [is\\_NxtHdrProto\\_Available](#)
- uint8\_t [NxtHdrProto](#)
- uint8\_t [is\\_IPv4Tos\\_Available](#)
- [unpack\\_qos\\_Tos\\_t](#) [IPv4Tos](#)
- uint8\_t [is\\_IPv6SrcAddr\\_Available](#)
- [unpack\\_qos\\_IPv6Addr\\_t](#) [IPv6SrcAddr](#)
- uint8\_t [is\\_IPv6DstAddr\\_Available](#)
- [unpack\\_qos\\_IPv6Addr\\_t](#) [IPv6DstAddr](#)
- uint8\_t [is\\_IPv6TrafCls\\_Available](#)
- [unpack\\_qos\\_IPv6TrafCls\\_t](#) [IPv6TrafCls](#)
- uint8\_t [is\\_IPv6Label\\_Available](#)
- uint32\_t [IPv6Label](#)
- uint8\_t [is\\_TCPSrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [TCPSrcPort](#)
- uint8\_t [is\\_TCPDstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [TCPDstPort](#)
- uint8\_t [is\\_UDPSrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [UDPSrcPort](#)
- uint8\_t [is\\_UDPDstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [UDPDstPort](#)
- uint8\_t [is\\_EspSpi\\_Available](#)
- uint32\_t [EspSpi](#)
- uint8\_t [is\\_Precedence\\_Available](#)
- uint16\_t [Precedence](#)
- uint8\_t [is\\_Id\\_Available](#)
- uint16\_t [Id](#)
- uint8\_t [is\\_TransrcPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [TransrcPort](#)
- uint8\_t [is\\_TransdstPort\\_Available](#)
- [unpack\\_qos\\_Port\\_t](#) [TransdstPort](#)

### 8.942.1 Detailed Description

This structure contains the QoS Filter Request.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

## Parameters

<i>index</i>	Mandatory parameter IP filter index Integer that uniquely identifies each filter instance This TLV must be present in the request
<i>version</i>	Mandatory parameter 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>IPv4SrcAddr</i>	IPv4 filter source address See <a href="#">unpack_qos_IPv4Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv4DstAddr</i>	IPv4 filter destination address See <a href="#">unpack_qos_IPv4Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>NxtHdrProto</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>IPv4Tos</i>	IPv4 filter type of service See <a href="#">unpack_qos_Tos_t</a> for more information
<i>IPv6SrcAddr</i>	IPv6 filter source address See <a href="#">unpack_qos_IPv6Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv6DstAddr</i>	IPv6 filter destination address See <a href="#">unpack_qos_IPv6Addr_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>IPv6TrafCls</i>	IPv6 filter traffic class See <a href="#">unpack_qos_IPv6TrafCls_t</a> for more information
<i>IPv6Label</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>TCPSrcPort</i>	TCP filter source port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>TCPDstPort</i>	TCP filter destination port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>UDPSrcPort</i>	UDP filter source port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>UDPDstPort</i>	UDP filter destination port filter See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>• Implemented only for unsolicited indication</li> </ul>
<i>EspSpi</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>Precedence</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>Id</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>TranSrcPort</i>	Transport protocolfilter source port See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>
<i>UDPDstPort</i>	Transport protocol filter destination port See <a href="#">unpack_qos_Port_t</a> for more information <ul style="list-style-type: none"> <li>Implemented only for unsolicited indication</li> </ul>

## 8.942.2 Field Documentation

8.942.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`

8.942.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`

8.942.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`

8.942.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`

8.942.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`

8.942.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`

8.942.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`

8.942.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`

8.942.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`

8.942.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`

8.942.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`

8.942.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`

8.942.2.13 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available`

8.942.2.14 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available`

8.942.2.15 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available`

8.942.2.16 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available`

8.942.2.17 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available`

8.942.2.18 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available`

8.942.2.19 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available`

8.942.2.20 `uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available`

8.942.2.21 `uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available`

8.942.2.22 `uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available`

8.942.2.23 `uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available`

8.942.2.24 `uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available`

8.942.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available`

8.942.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available`

8.942.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`

8.942.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`

8.942.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`

8.942.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort`

8.942.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort`

8.942.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`

8.942.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`

8.942.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort`

8.942.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`

8.942.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

## 8.943 `unpack_qos_swiQosFlow_t` Struct Reference

### Data Fields

- `uint8_t index`
- `uint8_t is_ProfileId3GPP2_Available`
- `uint16_t ProfileId3GPP2`
- `uint8_t is_val_3GPP2Pri_Available`
- `uint8_t val_3GPP2Pri`
- `uint8_t is_TrafficClass_Available`
- `uint8_t TrafficClass`
- `uint8_t is_DataRate_Available`
- `unpack_qos_dataRate_t DataRate`
- `uint8_t is_TokenBucket_Available`
- `unpack_qos_tokenBucket_t TokenBucket`
- `uint8_t is_Latency_Available`
- `uint32_t Latency`
- `uint8_t is_Jitter_Available`
- `uint32_t Jitter`
- `uint8_t is_PktErrRate_Available`
- `unpack_qos_pktErrRate_t PktErrRate`
- `uint8_t is_MinPolicedPktSz_Available`
- `uint32_t MinPolicedPktSz`
- `uint8_t is_MaxAllowedPktSz_Available`
- `uint32_t MaxAllowedPktSz`
- `uint8_t is_val_3GPPResResidualBER_Available`
- `uint16_t val_3GPPResResidualBER`

- [uint8\\_t is\\_val\\_3GPPTraHdlPri\\_Available](#)
- [uint8\\_t val\\_3GPPTraHdlPri](#)
- [uint8\\_t is\\_val\\_3GPPImCn\\_Available](#)
- [uint8\\_t val\\_3GPPImCn](#)
- [uint8\\_t is\\_val\\_3GPPSigInd\\_Available](#)
- [uint8\\_t val\\_3GPPSigInd](#)
- [uint8\\_t is\\_LteQci\\_Available](#)
- [uint8\\_t LteQci](#)

### 8.943.1 Detailed Description

This structure contains the QoS Flow Request.

Please check is\_<Param\_Name>\_Available field for presence of optional parameters

#### Parameters

<i>index</i>	<ul style="list-style-type: none"> <li>• Mandatory parameter</li> <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>ProfileId3GPP2</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>val_3GPP2Pri</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>TrafficClass</i>	<ul style="list-style-type: none"> <li>• IP flow traffic class</li> <li>• Integer that designates the requested traffic class: <ul style="list-style-type: none"> <li>• 0 – Conversational</li> <li>• 1 – Streaming</li> <li>• 2 – Interactive</li> <li>• 3 – Background</li> </ul> </li> </ul>
<i>DataRate</i>	<ul style="list-style-type: none"> <li>• IP flow data rate min max</li> <li>• See <a href="#">unpack_qos_dataRate_t</a> for more information</li> </ul>
<i>TokenBucket</i>	<ul style="list-style-type: none"> <li>• IP flow data rate token bucket</li> <li>• See <a href="#">unpack_qos_tokenBucket_t</a> for more information</li> </ul>
<i>Latency</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>Jitter</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>PktErrRate</i>	<ul style="list-style-type: none"> <li>• IP flow packet error rate</li> <li>• See <a href="#">unpack_qos_pktErrRate_t</a> for more information</li> </ul>
<i>MinPolicedPktSz</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>MaxAllowedPktSz</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>val_3GPPRes-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>val_3GPPTra-HdlPri</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>val_3GPPImCn</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>val_3GPPSigInd</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>LteQci</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>

## 8.943.2 Field Documentation

8.943.2.1 `unpack_qos_dataRate_t unpack_qos_swiQosFlow_t::DataRate`

8.943.2.2 `uint8_t unpack_qos_swiQosFlow_t::index`

8.943.2.3 `uint8_t unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.943.2.4 `uint8_t unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.943.2.5 `uint8_t unpack_qos_swiQosFlow_t::is_Latency_Available`

8.943.2.6 `uint8_t unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.943.2.7 `uint8_t unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.943.2.8 `uint8_t unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.943.2.9 `uint8_t unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.943.2.10 `uint8_t unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.943.2.11 `uint8_t unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.943.2.12 `uint8_t unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.943.2.13 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.943.2.14 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

8.943.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`

8.943.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available`

8.943.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`

8.943.2.18 `uint32_t unpack_qos_swiQosFlow_t::Jitter`

- 8.943.2.19 `uint32_t unpack_qos_swiQosFlow_t::Latency`
- 8.943.2.20 `uint8_t unpack_qos_swiQosFlow_t::LteQci`
- 8.943.2.21 `uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz`
- 8.943.2.22 `uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz`
- 8.943.2.23 `unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate`
- 8.943.2.24 `uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2`
- 8.943.2.25 `unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket`
- 8.943.2.26 `uint8_t unpack_qos_swiQosFlow_t::TrafficClass`
- 8.943.2.27 `uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri`
- 8.943.2.28 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPImCn`
- 8.943.2.29 `uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER`
- 8.943.2.30 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd`
- 8.943.2.31 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri`

## 8.944 `unpack_qos_tokenBucket_t` Struct Reference

### Data Fields

- `uint32_t` [peakRate](#)
- `uint32_t` [tokenRate](#)
- `uint32_t` [bucketSz](#)

### 8.944.1 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

### 8.944.2 Field Documentation

- 8.944.2.1 `uint32_t unpack_qos_tokenBucket_t::bucketSz`
- 8.944.2.2 `uint32_t unpack_qos_tokenBucket_t::peakRate`
- 8.944.2.3 `uint32_t unpack_qos_tokenBucket_t::tokenRate`

## 8.945 unpack\_qos\_Tos\_t Struct Reference

### Data Fields

- uint8\_t [val](#)
- uint8\_t [mask](#)

### 8.945.1 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>

### 8.945.2 Field Documentation

8.945.2.1 uint8\_t unpack\_qos\_Tos\_t::mask

8.945.2.2 uint8\_t unpack\_qos\_Tos\_t::val

## 8.946 unpack\_QosFlowStat\_t Struct Reference

### Data Fields

- uint32\_t [bearerId](#)
- uint32\_t [tx\\_pkt](#)
- uint32\_t [tx\\_pkt\\_drp](#)
- uint64\_t [tx\\_bytes](#)
- uint64\_t [tx\\_bytes\\_drp](#)

### 8.946.1 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>

## 8.946.2 Field Documentation

8.946.2.1 `uint32_t unpack_QosFlowStat_t::bearerId`

8.946.2.2 `uint64_t unpack_QosFlowStat_t::tx_bytes`

8.946.2.3 `uint64_t unpack_QosFlowStat_t::tx_bytes_drp`

8.946.2.4 `uint32_t unpack_QosFlowStat_t::tx_pkt`

8.946.2.5 `uint32_t unpack_QosFlowStat_t::tx_pkt_drp`

## 8.947 `unpack_sms_SendSMS_t` Struct Reference

### Data Fields

- `uint16_t` [messageID](#)
- `uint32_t` [messageFailureCode](#)

### 8.947.1 Detailed Description

#### Parameters

<i>messageID</i>	<ul style="list-style-type: none"> <li>• WMS message ID</li> </ul>
<i>messageFailureCode</i>	<ul style="list-style-type: none"> <li>• pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF</li> </ul>

## 8.947.2 Field Documentation

8.947.2.1 `uint32_t unpack_sms_SendSMS_t::messageFailureCode`

8.947.2.2 `uint16_t unpack_sms_SendSMS_t::messageID`

## 8.948 `unpack_sms_SetNewSMSCallback_ind_t` Struct Reference

### Data Fields

- `struct` [newMTMessageTlv](#) `NewMMTlv`
- `struct` [transferRouteMessageTlv](#) `TRMessageTlv`
- `struct` [messageModeTlv](#) `MMTlv`
- `struct` [sMSEtwsMessageTlv](#) `ETWSTlv`
- `struct` [eTWSPLMNInfoTlv](#) `ETWSPLMNTlv`

- struct [sMSCAddressTlv](#) [SMSTlv](#)
- struct [sMSONIMSTlv](#) [IMSTlv](#)

### 8.948.1 Detailed Description

#### Parameters

<i>NewMMTlv</i>	<ul style="list-style-type: none"> <li>• MT message</li> </ul>
<i>TRMessageTlv</i>	<ul style="list-style-type: none"> <li>• Transfer Route MT Message</li> <li>• See <a href="#">transferRouteMessageTlv</a> for more information</li> </ul>
<i>MMTlv</i>	<ul style="list-style-type: none"> <li>• Message mode</li> <li>• See <a href="#">messageModeTlv</a> for more information</li> </ul>
<i>ETWSTlv</i>	<ul style="list-style-type: none"> <li>• ETWS Message</li> <li>• See <a href="#">sMSEtwsMessageTlv</a> for more information</li> </ul>
<i>ETWSPLMNTlv</i>	<ul style="list-style-type: none"> <li>• ETWS PLMN Information</li> <li>• See <a href="#">eTWSPLMNInfoTlv</a> for more information</li> </ul>
<i>SMSTlv</i>	<ul style="list-style-type: none"> <li>• SMSC Address</li> <li>• See <a href="#">sMSCAddressTlv</a> for more information</li> </ul>
<i>IMSTlv</i>	<ul style="list-style-type: none"> <li>• SMS on IMS</li> <li>• See <a href="#">sMSONIMSTlv</a> for more information</li> </ul>

### 8.948.2 Field Documentation

8.948.2.1 struct [eTWSPLMNInfoTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv`

8.948.2.2 struct [sMSEtwsMessageTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv`

8.948.2.3 struct [sMSONIMSTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::IMSTlv`

8.948.2.4 struct [messageModeTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::MMTlv`

8.948.2.5 struct [newMTMessageTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv`

8.948.2.6 struct [sMSCAddressTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::SMSTlv`

8.948.2.7 struct [transferRouteMessageTlv](#) `unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv`

## 8.949 unpack\_sms\_SetNewSMSCallback\_t Struct Reference

## 8.950 unpack\_sms\_SLQSDeleteSMS\_t Struct Reference

## 8.951 unpack\_sms\_SLQSGetSMS\_t Struct Reference

### Data Fields

- uint32\_t [messageTag](#)
- uint32\_t [messageFormat](#)
- uint32\_t [messageSize](#)
- uint8\_t [message](#) [2048]

### 8.951.1 Detailed Description

#### Parameters

<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> <li>– 2 - Mobile originated and sent</li> <li>– 3 - Mobile originated but not yet sent</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>• Upon input the maximum number of bytes that can be written to the message array.</li> </ul>

- Upon successful output the actual number of bytes written to the message array.

#### Parameters

<i>message</i>	<ul style="list-style-type: none"> <li>• The message contents array</li> </ul>
----------------	--

### 8.951.2 Field Documentation

8.951.2.1 uint8\_t unpack\_sms\_SLQSGetSMS\_t::message[2048]

8.951.2.2 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageFormat

8.951.2.3 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageSize

8.951.2.4 uint32\_t unpack\_sms\_SLQSGetSMS\_t::messageTag

## 8.952 unpack\_sms\_SLQSGetSMSList\_t Struct Reference

## Data Fields

- uint32\_t [messageListSize](#)
- [qmiSmsMessageList](#) [messageList](#) [255]

### 8.952.1 Detailed Description

#### Parameters

<i>messageListSize</i>	<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>messageList</i>	<ul style="list-style-type: none"><li>• Message List</li><li>• See <a href="#">qmiSmsMessageList</a> for more information</li></ul>

### 8.952.2 Field Documentation

8.952.2.1 [qmiSmsMessageList](#) [unpack\\_sms\\_SLQSGetSMSList\\_t::messageList](#)[255]

8.952.2.2 [uint32\\_t](#) [unpack\\_sms\\_SLQSGetSMSList\\_t::messageListSize](#)

## 8.953 unpack\_sms\_SLQSMModifySMSStatus\_t Struct Reference

## 8.954 unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t Struct Reference

## Data Fields

- [uint8\\_t](#) [storageType](#)
- [uint8\\_t](#) [messageMode](#)

### 8.954.1 Detailed Description

#### 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>messageMode</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>

### 8.954.2 Field Documentation

8.954.2.1 [uint8\\_t](#) [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t::messageMode](#)

8.954.2.2 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType`

## 8.955 `unpack_swiloc_SwiLocGetAutoStart_t` Struct Reference

### Data Fields

- `uint8_t function`
- `int function_reported`
- `uint8_t fix_type`
- `int fix_type_reported`
- `uint8_t max_time`
- `int max_time_reported`
- `uint32_t max_dist`
- `int max_dist_reported`
- `uint32_t fix_rate`
- `int fix_rate_reported`

### 8.955.1 Detailed Description

This structure contains SWI LOC Get Auto Start setting

#### Parameters

<i>function</i>	<ul style="list-style-type: none"> <li>• Setting to indicate when modem should start an automatic GNSS fix <ul style="list-style-type: none"> <li>– 0 - disabled</li> <li>– 1 - At bootup</li> <li>– 2 - When NMEA port is opened</li> </ul> </li> </ul>
<i>function_reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>fix_type</i>	<ul style="list-style-type: none"> <li>• Type of GNSS fix: <ul style="list-style-type: none"> <li>– 1 - Default Engine mode</li> <li>– 2 - MS-Based</li> <li>– 3 - MS-Assisted</li> <li>– 4 - Standalone</li> </ul> </li> </ul>
<i>fix_type_reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>max_time</i>	<ul style="list-style-type: none"> <li>• Maximum time allowed for the receiver to get a fix in seconds</li> <li>• Valid range: 1-255</li> </ul>
<i>max_time_reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>

<i>max_dist</i>	<ul style="list-style-type: none"> <li>• Maximum uncertainty of a fix measured by distance in meters</li> <li>• Valid range: 1 - 4294967280</li> </ul>
<i>max_dist_-reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>
<i>fix_rate</i>	<ul style="list-style-type: none"> <li>• Time between fixes in seconds</li> <li>• Valid range: 1–65535</li> </ul>
<i>fix_rate_-reported</i>	<ul style="list-style-type: none"> <li>• 0 - not reported by modem</li> <li>• 1 - reported by modem</li> </ul>

## 8.955.2 Field Documentation

8.955.2.1 uint32\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_rate

8.955.2.2 int unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_rate\_reported

8.955.2.3 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_type

8.955.2.4 int unpack\_swiloc\_SwiLocGetAutoStart\_t::fix\_type\_reported

8.955.2.5 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::function

8.955.2.6 int unpack\_swiloc\_SwiLocGetAutoStart\_t::function\_reported

8.955.2.7 uint32\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_dist

8.955.2.8 int unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_dist\_reported

8.955.2.9 uint8\_t unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_time

8.955.2.10 int unpack\_swiloc\_SwiLocGetAutoStart\_t::max\_time\_reported

## 8.956 unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t Struct Reference

### Data Fields

- uint32\_t [eventType](#)
- [unpack\\_omaDmFotaTlv\\_t](#) SessionInfoFota
- [unpack\\_omaDmConfigTlv\\_t](#) SessionInfoConfig
- [unpack\\_omaDmNotificationsTlv\\_t](#) SessionInfoNotification

### 8.956.1 Detailed Description

Structure that contains OMA indication information based on eventType Structures for which the event is not valid will have values set to 0

## 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>SessionInfo-Fota[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmFotaTlv_t</a> for more information</li> </ul>
<i>SessionInfo-Config[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmConfigTlv_t</a> for more information</li> </ul>
<i>SessionInfo-Notification[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_omaDmNotificationsTlv_t</a> for more information</li> </ul>

## 8.956.2 Field Documentation

8.956.2.1 `uint32_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::eventType`

8.956.2.2 `unpack_omaDmConfigTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig`

8.956.2.3 `unpack_omaDmFotaTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota`

8.956.2.4 `unpack_omaDmNotificationsTlv_t unpack_swisma_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification`

8.957 `unpack_swisma_SLQSOMADMGetSessionInfo_t` Struct Reference

## Data Fields

- `uint8_t Status`
- `uint16_t UpdateCompleteStatus`
- `uint8_t Severity`
- `uint16_t SourceLength`
- `uint8_t Source [255]`
- `uint16_t PkgNameLength`
- `uint8_t PkgName [255]`
- `uint16_t PkgDescLength`
- `uint8_t PkgDescription [255]`
- `uint16_t DateLength`
- `uint8_t Date [255]`
- `uint16_t TimeLength`
- `uint8_t Time [255]`
- `uint8_t SessionType`
- `uint8_t SessionState`
- `uint16_t RetryCount`

## 8.957.1 Detailed Description

Structure that contains the session type for OMA get session info unpack command Also used as input parameter to specify the size of variable parameters. (ref. notes)

## Parameters

<i>Status</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>Update-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>Severity</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>SourceLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Vendor Name String in Bytes.</li> </ul>
<i>Source</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>PkgNameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.</li> </ul>
<i>PkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>PkgDescLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>PkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>DateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> </ul>
<i>Date</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>TimeLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Time String in Bytes.</li> </ul>

<i>Time</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII</li> <li>• See <a href="#">LIBPACK_MAX_SWIOMA_STR_LEN</a> for more information</li> </ul>
<i>SessionType</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>SessionState</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>RetryCount</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>

## 8.957.2 Field Documentation

8.957.2.1 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Date[255]`

8.957.2.2 `uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::DateLength`

8.957.2.3 `uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescLength`

8.957.2.4 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgDescription[255]`

8.957.2.5 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgName[255]`

8.957.2.6 `uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::PkgNameLength`

8.957.2.7 `uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::RetryCount`

8.957.2.8 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionState`

8.957.2.9 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SessionType`

8.957.2.10 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Severity`

8.957.2.11 `uint8_t unpack_swioma_SLQSOMADMGetSessionInfo_t::Source[255]`

8.957.2.12 `uint16_t unpack_swioma_SLQSOMADMGetSessionInfo_t::SourceLength`

8.957.2.13 uint8\_t unpack\_swisma\_SLQSOMADMGetSessionInfo\_t::Status

8.957.2.14 uint8\_t unpack\_swisma\_SLQSOMADMGetSessionInfo\_t::Time[255]

8.957.2.15 uint16\_t unpack\_swisma\_SLQSOMADMGetSessionInfo\_t::TimeLength

8.957.2.16 uint16\_t unpack\_swisma\_SLQSOMADMGetSessionInfo\_t::UpdateCompleteStatus

## 8.958 unpack\_swisma\_SLQSOMADMGetSettings\_t Struct Reference

### Data Fields

- uint32\_t [OMADMEEnabled](#)
- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t [Autosdm](#)
- uint8\_t [FwAutoCheck](#)

### 8.958.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

#### Parameters

<i>OMADM-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>FOTAdownload[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>FOTAUpdate[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>Autosdm[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>FwAutoCheck[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>

## 8.958.2 Field Documentation

8.958.2.1 `uint8_t unpack_swima_SLQSOMADMGetSettings_t::Autosdm`

8.958.2.2 `uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAdownload`

8.958.2.3 `uint8_t unpack_swima_SLQSOMADMGetSettings_t::FOTAupdate`

8.958.2.4 `uint8_t unpack_swima_SLQSOMADMGetSettings_t::FwAutoCheck`

8.958.2.5 `uint32_t unpack_swima_SLQSOMADMGetSettings_t::OMADMEabled`

## 8.959 unpack\_swima\_SLQSOMADMStartSession\_t Struct Reference

### Data Fields

- `uint32_t` [FwAvailability](#)

### 8.959.1 Detailed Description

Structure that contains the responses for OMA start session command

#### Parameters

<i>pFwAvailability[OUT]</i>	<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>
-----------------------------	---

## 8.959.2 Field Documentation

8.959.2.1 uint32\_t unpack\_swima\_SLQSOMADMStartSession\_t::FwAvailability

## 8.960 unpack\_uim\_ChangePin\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- uint32\_t \* [pIndicationToken](#)
- uint16\_t [Tlvresult](#)

### 8.960.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

#### Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</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 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.

### 8.960.2 Field Documentation

8.960.2.1 [uim\\_encryptedPIN1](#)\* [unpack\\_uim\\_ChangePin\\_t::pEncryptedPIN1](#)

8.960.2.2 uint32\_t\* [unpack\\_uim\\_ChangePin\\_t::pIndicationToken](#)

8.960.2.3 [uim\\_remainingRetries](#)\* [unpack\\_uim\\_ChangePin\\_t::pRemainingRetries](#)

8.960.2.4 uint16\_t [unpack\\_uim\\_ChangePin\\_t::Tlvresult](#)

## 8.961 unpack\_uim\_GetCardStatus\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)
- [uim\\_hotSwapStatus](#) \* [pHotSwapStatus](#)
- uint16\_t [Tlvresult](#)

### 8.961.1 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="#">uim_cardStatus</a> for more information.</li> </ul>
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_hotSwapStatus</a> for more information.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not returned.

### 8.961.2 Field Documentation

8.961.2.1 [uim\\_cardStatus](#)\* [unpack\\_uim\\_GetCardStatus\\_t::pCardStatus](#)

8.961.2.2 [uim\\_hotSwapStatus](#)\* [unpack\\_uim\\_GetCardStatus\\_t::pHotSwapStatus](#)

8.961.2.3 [uint16\\_t](#) [unpack\\_uim\\_GetCardStatus\\_t::Tlvresult](#)

## 8.962 [unpack\\_uim\\_ReadTransparent\\_t](#) Struct Reference

#### Data Fields

- [uim\\_cardResult](#) \* [pCardResult](#)
- [uim\\_readResult](#) \* [pReadResult](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint8\\_t](#) \* [pEncryptedData](#)
- [uint16\\_t](#) [Tlvresult](#)

### 8.962.1 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 <a href="#">read_result</a> is encrypted.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.962.2 Field Documentation

8.962.2.1 `uim_cardResult*` `unpack_uim_ReadTransparent_t::pCardResult`

8.962.2.2 `uint8_t*` `unpack_uim_ReadTransparent_t::pEncryptedData`

8.962.2.3 `uint32_t*` `unpack_uim_ReadTransparent_t::pIndicationToken`

8.962.2.4 `uim_readResult*` `unpack_uim_ReadTransparent_t::pReadResult`

8.962.2.5 `uint16_t` `unpack_uim_ReadTransparent_t::Tlvresult`

## 8.963 unpack\_uim\_SetPinProtection\_t Struct Reference

## Data Fields

- `uim_remainingRetries` \* `pRemainingRetries`
- `uim_encryptedPIN1` \* `pEncryptedPIN1`
- `uint32_t` \* `pIndicationToken`
- `uint16_t` `Tlvresult`

## 8.963.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

## Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</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 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.

## 8.963.2 Field Documentation

8.963.2.1 `uim_encryptedPIN1*` `unpack_uim_SetPinProtection_t::pEncryptedPIN1`

8.963.2.2 `uint32_t*` `unpack_uim_SetPinProtection_t::pIndicationToken`

8.963.2.3 `uim_remainingRetries*` `unpack_uim_SetPinProtection_t::pRemainingRetries`

8.963.2.4 uint16\_t unpack\_uim\_SetPinProtection\_t::Tlvresult

## 8.964 unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t Struct Reference

### Data Fields

- [slots\\_t slotsstatusChange](#)
- uint8\_t [bNumberOfPhySlots](#)

### 8.964.1 Detailed Description

Structure consist of card status params

#### Parameters

<i>slotsstatus- Change</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slot_t</a> for more information</li> </ul>
<i>bNumberOfPhy- Slots</i>	<ul style="list-style-type: none"> <li>• Number of Physical Slot(s)</li> </ul>

### 8.964.2 Field Documentation

8.964.2.1 uint8\_t unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t::bNumberOfPhySlots

8.964.2.2 slots\_t unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t::slotsstatusChange

## 8.965 unpack\_uim\_SLQSUIEventRegister\_t Struct Reference

### Data Fields

- uint32\_t [eventMask](#)

### 8.965.1 Detailed Description

#### Parameters

<i>eventMask</i>	- bit 0 - card status <ul style="list-style-type: none"> <li>• bit 1 - SAP connection</li> <li>• bit 4 - physical slot status</li> </ul>
------------------	--

### 8.965.2 Field Documentation

8.965.2.1 uint32\_t unpack\_uim\_SLQSUIEventRegister\_t::eventMask

## 8.966 unpack\_uim\_SLQSUIGetSlotsStatus\_t Struct Reference

### Data Fields

- uint8\_t \* [pNumberOfPhySlot](#)

- [slots\\_t](#) \* [pUimSlotsStatus](#)

### 8.966.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>pNumberOfPhy-Slot</i>	<ul style="list-style-type: none"> <li>• Number of sets of the Slot Status.</li> </ul>
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> <li>• Slots Status See <a href="#">slots_t</a> for more information..</li> </ul>

### 8.966.2 Field Documentation

8.966.2.1 `uint8_t* unpack_uim_SLQSUIGetSlotsStatus_t::pNumberOfPhySlot`

8.966.2.2 `slots_t* unpack_uim_SLQSUIGetSlotsStatus_t::pUimSlotsStatus`

## 8.967 unpack\_uim\_SLQSUISetStatusChangeCallBack\_ind\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)

### 8.967.1 Detailed Description

This structure contains information about Status change callback.

#### Parameters

<i>pCardStatus</i>	Card Status <ul style="list-style-type: none"> <li>• See <a href="#">uim_cardStatus</a> for more information.</li> </ul>
--------------------	--

### 8.967.2 Field Documentation

8.967.2.1 `uim_cardStatus* unpack_uim_SLQSUISetStatusChangeCallBack_ind_t::pCardStatus`

## 8.968 unpack\_uim\_UnblockPin\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- `uint32_t` \* [pIndicationToken](#)
- `uint16_t` [Tlvresult](#)

### 8.968.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

#### Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</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 is provided in a subsequent indication.</li> <li>0xFFFFFFFF, if unavailable</li> </ul>

### 8.968.2 Field Documentation

8.968.2.1 `uim_encryptedPIN1*` `unpack_uim_UnblockPin_t::pEncryptedPIN1`

8.968.2.2 `uint32_t*` `unpack_uim_UnblockPin_t::pIndicationToken`

8.968.2.3 `uim_remainingRetries*` `unpack_uim_UnblockPin_t::pRemainingRetries`

8.968.2.4 `uint16_t` `unpack_uim_UnblockPin_t::Tlvresult`

## 8.969 `unpack_uim_VerifyPin_t` Struct Reference

#### Data Fields

- `uim_remainingRetries` \* `pRemainingRetries`
- `uim_encryptedPIN1` \* `pEncryptedPIN1`
- `uint32_t` \* `pIndicationToken`
- `uint16_t` `Tlvresult`

### 8.969.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

#### Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</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 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.

## 8.969.2 Field Documentation

8.969.2.1 uim\_encryptedPIN1\* unpack\_uim\_VerifyPin\_t::pEncryptedPIN1

8.969.2.2 uint32\_t\* unpack\_uim\_VerifyPin\_t::pIndicationToken

8.969.2.3 uim\_remainingRetries\* unpack\_uim\_VerifyPin\_t::pRemainingRetries

8.969.2.4 uint16\_t unpack\_uim\_VerifyPin\_t::Tlvresult

## 8.970 unpack\_wds\_GetConnectionRate\_t Struct Reference

## Data Fields

- uint32\_t [currentChannelTXRate](#)
- uint32\_t [currentChannelRXRate](#)
- uint32\_t [maxChannelTXRate](#)
- uint32\_t [maxChannelRXRate](#)

## 8.970.1 Detailed Description

## Parameters

<i>currentChannel-TXRate</i>	Instantaneous channel Tx rate
<i>currentChannel-RXRate</i>	Instantaneous channel Rx rate
<i>maxChannelTX-Rate</i>	Maximum Tx rate
<i>maxChannelRX-Rate</i>	Maximum Rx rate

## 8.970.2 Field Documentation

8.970.2.1 uint32\_t unpack\_wds\_GetConnectionRate\_t::currentChannelRXRate

8.970.2.2 uint32\_t unpack\_wds\_GetConnectionRate\_t::currentChannelTXRate

8.970.2.3 uint32\_t unpack\_wds\_GetConnectionRate\_t::maxChannelRXRate

8.970.2.4 uint32\_t unpack\_wds\_GetConnectionRate\_t::maxChannelTXRate

## 8.971 unpack\_wds\_GetDefaultProfile\_t Struct Reference

## Data Fields

- uint32\_t [pdptype](#)
- uint32\_t [ipaddr](#)
- uint32\_t [pridns](#)
- uint32\_t [secdns](#)

- uint16\_t [ipaddrv6](#)
- uint16\_t [pridnsv6](#)
- uint16\_t [secdnsv6](#)
- uint32\_t [auth](#)
- uint8\_t [namesize](#)
- int8\_t [name](#) [255]
- uint8\_t [apnsize](#)
- int8\_t [apnname](#) [255]
- uint8\_t [usersize](#)
- int8\_t [username](#) [255]

### 8.971.1 Detailed Description

#### Parameters

<i>pdptype</i>	pdp type
<i>ipaddr</i>	ip address
<i>pridns</i>	primary dns
<i>secdns</i>	secondry dns
<i>ipaddrv6</i>	ip address v6
<i>pridnsv6</i>	primary dns v6
<i>secdnsv6</i>	secondry dns v6
<i>namesize</i>	profile name size
<i>name</i>	profile name
<i>apnsize</i>	apn size
<i>apnname</i>	apn name
<i>usersize</i>	username size
<i>username</i>	username

### 8.971.2 Field Documentation

8.971.2.1 int8\_t [unpack\\_wds\\_GetDefaultProfile\\_t::apnname](#)[255]

8.971.2.2 uint8\_t [unpack\\_wds\\_GetDefaultProfile\\_t::apnsize](#)

8.971.2.3 uint32\_t [unpack\\_wds\\_GetDefaultProfile\\_t::auth](#)

8.971.2.4 uint32\_t [unpack\\_wds\\_GetDefaultProfile\\_t::ipaddr](#)

8.971.2.5 uint16\_t [unpack\\_wds\\_GetDefaultProfile\\_t::ipaddrv6](#)

8.971.2.6 int8\_t [unpack\\_wds\\_GetDefaultProfile\\_t::name](#)[255]

8.971.2.7 uint8\_t [unpack\\_wds\\_GetDefaultProfile\\_t::namesize](#)

8.971.2.8 uint32\_t [unpack\\_wds\\_GetDefaultProfile\\_t::pdptype](#)

8.971.2.9 uint32\_t [unpack\\_wds\\_GetDefaultProfile\\_t::pridns](#)

8.971.2.10 uint16\_t [unpack\\_wds\\_GetDefaultProfile\\_t::pridnsv6](#)

8.971.2.11 uint32\_t [unpack\\_wds\\_GetDefaultProfile\\_t::secdns](#)

8.971.2.12 uint16\_t [unpack\\_wds\\_GetDefaultProfile\\_t::secdnsv6](#)

8.971.2.13 int8\_t unpack\_wds\_GetDefaultProfile\_t::username[255]

8.971.2.14 uint8\_t unpack\_wds\_GetDefaultProfile\_t::usersize

## 8.972 unpack\_wds\_GetDefaultProfileNum\_t Struct Reference

### Data Fields

- uint8\_t [index](#)

#### 8.972.1 Detailed Description

##### Parameters

<i>index</i>	profile index
--------------	---------------

#### 8.972.2 Field Documentation

8.972.2.1 uint8\_t unpack\_wds\_GetDefaultProfileNum\_t::index

## 8.973 unpack\_wds\_GetDormancyState\_t Struct Reference

### Data Fields

- uint32\_t [dormancyState](#)

#### 8.973.1 Detailed Description

##### Parameters

<i>dormancyState</i>	dormancy status
----------------------	-----------------

#### 8.973.2 Field Documentation

8.973.2.1 uint32\_t unpack\_wds\_GetDormancyState\_t::dormancyState

## 8.974 unpack\_wds\_GetLastMobileLError\_t Struct Reference

### Data Fields

- uint32\_t [error](#)

#### 8.974.1 Detailed Description

##### Parameters

<i>error</i>	last mip status 0-success >0- error code
--------------	--

#### 8.974.2 Field Documentation

8.974.2.1 uint32\_t unpack\_wds\_GetLastMobileLError\_t::error

## 8.975 unpack\_wds\_GetMobileIP\_t Struct Reference

### Data Fields

- uint32\_t [mipMode](#)

### 8.975.1 Detailed Description

#### Parameters

<i>mipMode</i>	mobile IP mode
----------------	----------------

### 8.975.2 Field Documentation

8.975.2.1 uint32\_t unpack\_wds\_GetMobileIP\_t::mipMode

## 8.976 unpack\_wds\_GetMobileIPProfile\_t Struct Reference

### Data Fields

- uint8\_t [enabled](#)
- uint32\_t [address](#)
- uint32\_t [primaryHA](#)
- uint32\_t [secondaryHA](#)
- uint8\_t [revTunneling](#)
- uint8\_t [naiSize](#)
- int8\_t [NAI](#) [255]
- uint32\_t [HASPI](#)
- uint32\_t [AAASPI](#)
- uint32\_t [HASState](#)
- uint32\_t [AAASState](#)

### 8.976.1 Detailed Description

#### Parameters

<i>enabled</i>	mobile ip profile state enabled/disabled
<i>address</i>	mobile ip profile home address
<i>primaryHA</i>	mobile ip profile home agent primary
<i>secondaryHA</i>	mobile ip profile secondary home agent address
<i>revTunneling</i>	mobile ip profile rev tunneling
<i>naiSize</i>	mobile ip profile NAI size
<i>NAI</i>	NAI string in ASCII text.
<i>HASPI</i>	HA security parameter index.
<i>AAASPI</i>	AAA server security parameter index.
<i>HASState</i>	Mobile IP Profile HA Key State
<i>AAASState</i>	Mobile IP Profile AAA Key State

### 8.976.2 Field Documentation

8.976.2.1 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::AAASPI

- 8.976.2.2 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::AAState
- 8.976.2.3 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::address
- 8.976.2.4 uint8\_t unpack\_wds\_GetMobileIPProfile\_t::enabled
- 8.976.2.5 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::HASPI
- 8.976.2.6 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::HASState
- 8.976.2.7 int8\_t unpack\_wds\_GetMobileIPProfile\_t::NAI[255]
- 8.976.2.8 uint8\_t unpack\_wds\_GetMobileIPProfile\_t::naiSize
- 8.976.2.9 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::primaryHA
- 8.976.2.10 uint8\_t unpack\_wds\_GetMobileIPProfile\_t::revTunneling
- 8.976.2.11 uint32\_t unpack\_wds\_GetMobileIPProfile\_t::secondaryHA

## 8.977 unpack\_wds\_GetPacketStatus\_t Struct Reference

### Data Fields

- uint32\_t tXPacketSuccesses
- uint32\_t rXPacketSuccesses
- uint32\_t tXPacketErrors
- uint32\_t rXPacketErrors
- uint32\_t tXPacketOverflows
- uint32\_t rXPacketOverflows
- uint64\_t tXOkBytesCount
- uint64\_t rXOkBytesCount
- uint64\_t tXOKBytesLastCall
- uint64\_t rXOKBytesLastCall
- uint32\_t tXDroppedCount
- uint32\_t rXDroppedCount

### 8.977.1 Detailed Description

#### Parameters

<i>tXPacket-Successes</i>	Tx Packets OK
<i>rXPacket-Successes</i>	Rx Packets OK
<i>tXPacketErrors</i>	Tx Packet Errors
<i>rXPacketErrors</i>	Rx Packet Errors
<i>tXPacket-Overflows</i>	Tx Overflows
<i>rXPacket-Overflows</i>	Rx Overflows
<i>tXOkBytesCount</i>	Tx Bytes OK
<i>rXOkBytesCount</i>	Rx Bytes OK
<i>tXOKBytesLast-Call</i>	Last call Tx Bytes OK

<i>rXOKBytesLast-Call</i>	Last call Rx Bytes OK
<i>tXDroppedCount</i>	Tx Packets Dropped
<i>rXDroppedCount</i>	Rx Packets Dropped

## 8.977.2 Field Documentation

8.977.2.1 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`

8.977.2.2 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount`

8.977.2.3 `uint64_t unpack_wds_GetPacketStatus_t::rXOKBytesLastCall`

8.977.2.4 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors`

8.977.2.5 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows`

8.977.2.6 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses`

8.977.2.7 `uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount`

8.977.2.8 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount`

8.977.2.9 `uint64_t unpack_wds_GetPacketStatus_t::tXOKBytesLastCall`

8.977.2.10 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors`

8.977.2.11 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows`

8.977.2.12 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses`

## 8.978 `unpack_wds_GetSessionDuration_t` Struct Reference

### Data Fields

- `uint64_t` [callDuration](#)

### 8.978.1 Detailed Description

#### Parameters

<i>callDuration</i>	call duration in milliseconds
---------------------	-------------------------------

## 8.978.2 Field Documentation

8.978.2.1 `uint64_t unpack_wds_GetSessionDuration_t::callDuration`

## 8.979 `unpack_wds_GetSessionState_t` Struct Reference

### Data Fields

- `uint32_t` [connectionStatus](#)

### 8.979.1 Detailed Description

#### Parameters

<a href="#">connection-Status</a>	state of the current packet data session
-----------------------------------	--

### 8.979.2 Field Documentation

8.979.2.1 `uint32_t unpack_wds_GetSessionState_t::connectionStatus`

## 8.980 unpack\_wds\_RMSetTransferStatistics\_t Struct Reference

### 8.981 unpack\_wds\_SetMobileIPProfile\_t Struct Reference

### 8.982 unpack\_wds\_SLQSCreateProfile\_t Struct Reference

#### Data Fields

- [PackCreateProfileOut](#) \* [pCreateProfileOut](#)
- `uint8_t` \* [pProfileID](#)
- `uint16_t` [Tlvresult](#)

### 8.982.1 Detailed Description

#### Parameters

<i>profile</i>	type
<i>profile</i>	index
<i>extended</i>	error

### 8.982.2 Field Documentation

8.982.2.1 `PackCreateProfileOut*` `unpack_wds_SLQSCreateProfile_t::pCreateProfileOut`

8.982.2.2 `uint8_t*` `unpack_wds_SLQSCreateProfile_t::pProfileID`

8.982.2.3 `uint16_t` `unpack_wds_SLQSCreateProfile_t::Tlvresult`

## 8.983 unpack\_wds\_SLQSDestroyProfile\_t Struct Reference

#### Data Fields

- `uint16_t` [extendedErrorCode](#)

### 8.983.1 Detailed Description

#### Parameters

<i>extendedError-Code</i>	extended error code
---------------------------	---------------------

## 8.983.2 Field Documentation

8.983.2.1 uint16\_t unpack\_wds\_SLQSDeleteProfile\_t::extendedErrorCode

## 8.984 unpack\_wds\_SLQSGet3GPPConfigItem\_t Struct Reference

### Data Fields

- uint16\_t [profileList](#) [5]
- uint8\_t [defaultPDNEnabled](#)
- uint8\_t [\\_3gppRelease](#)
- uint16\_t [LTEAttachProfileList](#) [24]
- uint16\_t [LTEAttachProfileListLen](#)

### 8.984.1 Detailed Description

#### Parameters

	<i>profileList</i>	Profile List
out	<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>• 0 - disabled</li> <li>• 1 - enabled</li> </ul>
out	<i>_3gppRelease</i>	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> <li>• 5 - Release_9 (In 9x30 and towerads)</li> <li>• 6 - Release_10 (In 9x30 and towerads)</li> <li>• 7 - Release_11 (In 9x30 and towerads)</li> </ul>
out	<i>LTEAttach-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> <li>– Please provide attach profiles in order of decreasing priority in this list.</li> </ul> </li> </ul>
in, out	<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: 1-24</li> <li>– This setting is only supported for MC/EM74xx onwards</li> </ul> </li> </ul>

## 8.984.2 Field Documentation

8.984.2.1 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::\_3gppRelease

8.984.2.2 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::defaultPDNEnabled

8.984.2.3 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::LTEAttachProfileList[24]

8.984.2.4 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::LTEAttachProfileListLen

8.984.2.5 uint16\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::profileList[5]

## 8.985 unpack\_wds\_SLQSGetCurrDataSystemStat\_t Struct Reference

### Data Fields

- uint8\_t [prefNetwork](#)
- uint8\_t [networkInfoLen](#)
- [currNetworkInfo](#) [currNetworkInfo](#) [255]

### 8.985.1 Detailed Description

#### Parameters

<a href="#">prefNetwork</a>	preferred network
<a href="#">networkInfoLen</a>	number of set of <a href="#">currNetworkInfo</a> elements
<a href="#">currNetworkInfo</a>	current network information.

### 8.985.2 Field Documentation

8.985.2.1 [currNetworkInfo](#) [unpack\\_wds\\_SLQSGetCurrDataSystemStat\\_t::currNetworkInfo](#)[255]

8.985.2.2 uint8\_t [unpack\\_wds\\_SLQSGetCurrDataSystemStat\\_t::networkInfoLen](#)

8.985.2.3 uint8\_t [unpack\\_wds\\_SLQSGetCurrDataSystemStat\\_t::prefNetwork](#)

## 8.986 unpack\_wds\_SLQSGetDataBearerTechnology\_t Struct Reference

### Data Fields

- uint8\_t [dataBearerMask](#)
- [qmiWSDDataBearerTechnology](#) [curDataBearerTechnology](#)
- [qmiWSDDataBearerTechnology](#) [lastCallDataBearerTechnology](#)

### 8.986.1 Detailed Description

#### Parameters

<a href="#">dataBearerMask</a>	bit mask indicates bearer info is for current and/or last call
<a href="#">curDataBearerTechnology</a>	current data bearer technology value
<a href="#">lastCallDataBearerTechnology</a>	last call data bearer technology value

### 8.986.2 Field Documentation

8.986.2.1 [qmiWSDDataBearerTechnology](#) [unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t::curDataBearerTechnology](#)

8.986.2.2 `uint8_t unpack_wds_SLQSGetDataBearerTechnology_t::dataBearerMask`

8.986.2.3 `qmiWSDDataBearerTechnology unpack_wds_SLQSGetDataBearerTechnology_t::lastCallDataBearerTechnology`

## 8.987 `unpack_wds_SLQSGetDUNCallInfo_t` Struct Reference

### Data Fields

- [connectionStatus](#) `connectionStatus`
- `uint16_t` [callEndReason](#)
- `uint64_t` [txOKBytesCount](#)
- `uint64_t` [rxOKBytesCount](#)
- `uint8_t` [dormancyStatus](#)
- `uint8_t` [dataBearerTech](#)
- [dunchannelRate](#) `channelRate`
- `uint64_t` [lastCallTXOKBytesCnt](#)
- `uint64_t` [lastCallRXOKBytesCnt](#)
- `uint64_t` [mdmCallDurationActive](#)
- `uint8_t` [lastCallDataBearerTech](#)

### 8.987.1 Detailed Description

#### Parameters

<a href="#">connectionStatus</a>	Connection Status
<a href="#">callEndReason</a>	Last Modem Call End Reason
<a href="#">txOKBytesCount</a>	Tx Bytes OK
<a href="#">rxOKBytesCount</a>	Rx Bytes OK
<a href="#">dormancyStatus</a>	Dormancy Status
<a href="#">dataBearerTech</a>	data bearer technology
<a href="#">channelRate</a>	data Channel Rate
<a href="#">lastCallTXOKBytesCnt</a>	Last Call Tx Bytes OK
<a href="#">lastCallRXOKBytesCnt</a>	Last Call Rx Bytes OK
<a href="#">mdmCallDurationActive</a>	Call active duration
<a href="#">lastCallDataBearerTech</a>	Last Call Data Bearer Technology

### 8.987.2 Field Documentation

8.987.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`

8.987.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`

8.987.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`

8.987.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`

8.987.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`

- 8.987.2.6    uint8\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::lastCallDataBearerTech
- 8.987.2.7    uint64\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::lastCallRXOKBytesCnt
- 8.987.2.8    uint64\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::lastCallTXOKBytesCnt
- 8.987.2.9    uint64\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::mdmCallDurationActive
- 8.987.2.10    uint64\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::rxOKBytesCount
- 8.987.2.11    uint64\_t unpack\_wds\_SLQSGetDUNCallInfo\_t::txOKBytesCount

## 8.988    unpack\_wds\_SLQSGetProfileSettings\_t Struct Reference

### Data Fields

- [UnPackGetProfileSettingOut](#) \* pProfileSettings
- uint8\_t [ProfileType](#)
- uint16\_t [Tlvresult](#)

### 8.988.1    Field Documentation

- 8.988.1.1    [UnPackGetProfileSettingOut](#)\* unpack\_wds\_SLQSGetProfileSettings\_t::pProfileSettings
- 8.988.1.2    uint8\_t unpack\_wds\_SLQSGetProfileSettings\_t::ProfileType
- 8.988.1.3    uint16\_t unpack\_wds\_SLQSGetProfileSettings\_t::Tlvresult

## 8.989    unpack\_wds\_SLQSGetRuntimeSettings\_t Struct Reference

### Data Fields

- uint32\_t [IPv4](#)
- uint8\_t [ProfileName](#) [128]
- uint32\_t [PDPTType](#)
- uint8\_t [APNName](#) [128]
- uint32\_t [PrimaryDNSV4](#)
- uint32\_t [SecondaryDNSV4](#)
- [LibPackUMTSQoS](#) UMTSGrantedQoS
- struct [wds\\_GPRSQoS](#) GPRSGrantedQoS
- uint8\_t [Username](#) [128]
- uint32\_t [Authentication](#)
- struct [wds\\_ProfileIdentifier](#) ProfileID
- uint32\_t [GWAddressV4](#)
- uint32\_t [SubnetMaskV4](#)
- uint8\_t [PCSCFAddrPCO](#)
- struct [wds\\_PCSCFIPv4ServerAddressList](#) ServerAddrList
- struct [wds\\_PCSCFFQDNAddressList](#) PCSCFFQDNAddrList
- uint16\_t [PrimaryDNSV6](#) [8]
- uint16\_t [SecondaryDNSV6](#) [8]
- uint32\_t [Mtu](#)
- struct [wds\\_DomainNameList](#) DomainList

- uint8\_t [IPFamilyPreference](#)
- uint8\_t [IMCNflag](#)
- uint16\_t [Technology](#)
- struct [wds\\_IPV6AddressInfo](#) [IPV6AddrInfo](#)
- struct [wds\\_IPV6GWAddressInfo](#) [IPV6GWAddrInfo](#)

### 8.989.1 Detailed Description

#### Parameters

<i>IPv4</i>	ipv4 address
<i>ProfileName</i>	profile name
<i>PDPTType</i>	PDP type
<i>APNName</i>	APN name
<i>PrimaryDNSV4</i>	
<i>SecondaryDNS-V4</i>	
<i>UMTSGranted-QoS</i>	UMTS Granted Qos
<i>GPRSGranted-QoS</i>	GPRS Granted QoS
<i>Username</i>	
<i>Authentication</i>	
<i>ProfielID</i>	
<i>GWAddressV4</i>	Gateway IPv4
<i>SubnetMaskV4</i>	Subnet mask IPV4
<i>PCSCFAddrPC-O</i>	
<i>PrimaryDNSV6</i>	Primary DNS IPV6
<i>SecondaryDNS-V6</i>	Secondary DNS IPV6
<i>UMTSGranted-QoS</i>	UMTS Granted Qos
<i>SecondaryDNS-V4</i>	
<i>Mtu</i>	Maximum Transfer Unit
<i>DomainList</i>	
<i>IPFamily-Preference</i>	

### 8.989.2 Field Documentation

8.989.2.1 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::APNName[128]

8.989.2.2 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::Authentication

8.989.2.3 struct wds\_DomainNameList unpack\_wds\_SLQSGetRuntimeSettings\_t::DomainList

8.989.2.4 struct wds\_GPRSQoS unpack\_wds\_SLQSGetRuntimeSettings\_t::GPRSGrantedQoS

8.989.2.5 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::GWAddressV4

8.989.2.6 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::IMCNflag

8.989.2.7 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::IPFamilyPreference

- 8.989.2.8 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::IPv4
- 8.989.2.9 struct wds\_IPV6AddressInfo unpack\_wds\_SLQSGetRuntimeSettings\_t::IPv6AddrInfo
- 8.989.2.10 struct wds\_IPV6GWAddressInfo unpack\_wds\_SLQSGetRuntimeSettings\_t::IPv6GWAddrInfo
- 8.989.2.11 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::Mtu
- 8.989.2.12 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::PCSCFAddrPCO
- 8.989.2.13 struct wds\_PCSCFFQDNAddressList unpack\_wds\_SLQSGetRuntimeSettings\_t::PCSCFFQDNAddrList
- 8.989.2.14 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::PDPTType
- 8.989.2.15 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::PrimaryDNSV4
- 8.989.2.16 uint16\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::PrimaryDNSV6[8]
- 8.989.2.17 struct wds\_ProfileIdentifier unpack\_wds\_SLQSGetRuntimeSettings\_t::ProfileID
- 8.989.2.18 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::ProfileName[128]
- 8.989.2.19 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::SecondaryDNSV4
- 8.989.2.20 uint16\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::SecondaryDNSV6[8]
- 8.989.2.21 struct wds\_PCSCFIPv4ServerAddressList unpack\_wds\_SLQSGetRuntimeSettings\_t::ServerAddrList
- 8.989.2.22 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::SubnetMaskV4
- 8.989.2.23 uint16\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::Technology
- 8.989.2.24 LibPackUMTSQoS unpack\_wds\_SLQSGetRuntimeSettings\_t::UMTSGrantedQoS
- 8.989.2.25 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::Username[128]

## 8.990 unpack\_wds\_SLQSModifyProfile\_t Struct Reference

### Data Fields

- uint16\_t \* [pExtErrorCode](#)

### 8.990.1 Detailed Description

#### Parameters

<i>extended</i>	error
-----------------	-------

### 8.990.2 Field Documentation

- 8.990.2.1 uint16\_t\* unpack\_wds\_SLQSModifyProfile\_t::pExtErrorCode

## 8.991 unpack\_wds\_SLQSSetIPFamilyPreference\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.991.1 Detailed Description

#### Parameters

<i>Tlvresult</i>	unpack result
------------------	---------------

### 8.991.2 Field Documentation

8.991.2.1 uint16\_t unpack\_wds\_SLQSSetIPFamilyPreference\_t::Tlvresult

## 8.992 unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t Struct Reference

## Data Fields

- uint8\_t [conn\\_status](#)
- uint8\_t [reconfigReqd](#)
- uint16\_t [sessionEndReason](#)
- uint16\_t [verboseSessnEndReasonType](#)
- uint16\_t [verboseSessnEndReason](#)
- uint8\_t [ipFamily](#)
- uint16\_t [techName](#)
- uint8\_t [bearerID](#)

### 8.992.1 Detailed Description

#### Parameters

<i>conn_status</i>	connection status
<i>reconfigReqd</i>	Indicates whether the network interface on the host needs to be reconfigured.
<i>sessionEndReason</i>	Call End Reason
<i>verboseSessnEndReasonType</i>	Verbose call end reason type
<i>verboseSessnEndReason</i>	Reason the call ended (verbose)
<i>ipFamily</i>	IP family of the packet data connection.
<i>techName</i>	Technology name of the packet data connection.
<i>bearerID</i>	<ul style="list-style-type: none"> <li>• bearer ID (3GPP) or RLP ID (3GPP2) of the packet data connection.</li> <li>• Valid Values - 0 to 16</li> <li>• 0xFF - Invalid value.</li> </ul>

### 8.992.2 Field Documentation

8.992.2.1 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::bearerID

8.992.2.2 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::conn\_status

8.992.2.3 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::ipFamily

8.992.2.4 uint8\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::reconfigReqd

8.992.2.5 uint16\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::sessionEndReason

8.992.2.6 uint16\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::techName

8.992.2.7 uint16\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::verboseSessnEndReason

8.992.2.8 uint16\_t unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t::verboseSessnEndReasonType

## 8.993 unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [xferStatAvail](#)
- uint64\_t [tx\\_bytes](#)
- uint64\_t [rx\\_bytes](#)
- uint64\_t [tx\\_pkts](#)
- uint64\_t [rx\\_pkts](#)
- uint8\_t [mipstatAvail](#)
- uint32\_t [mipStatus](#)
- uint8\_t [dBTechAvail](#)
- uint32\_t [dBTechnology](#)
- uint8\_t [dormancyStatAvail](#)
- uint32\_t [dormancyStatus](#)
- uint8\_t [currDBTechAvail](#)
- uint32\_t [ratMask](#)
- uint32\_t [soMask](#)
- uint8\_t [dataSysStatAvail](#)
- uint8\_t [prefNetwork](#)
- uint8\_t [netInfoLen](#)
- [wds\\_currNetworkInfo](#) [currNWInfo](#) [255]

### 8.993.1 Detailed Description

#### Parameters

<i>xferStatAvail</i>	transfer statistic available
<i>tx_bytes</i>	transmit bytes
<i>rx_bytes</i>	received bytes
<i>tx_pkts</i>	transmit packets
<i>rx_pkts</i>	received packets
<i>mipstatAvail</i>	Mobile IP status available
<i>mipStatus</i>	Mobile IP status
<i>dBTechAvail</i>	Data Bearer technology available
<i>dBTechnology</i>	Data Bearer technology
<i>dormancyStatAvail</i>	Dormancy status available
<i>dormancyStatus</i>	Dormancy status
<i>currDBTechAvail</i>	Current Data Bearer technology available
<i>ratMask</i>	RAT mask to indicate type of technology
<i>soMask</i>	SO mask to indicate the service type
<i>dataSysStatAvail</i>	Data System Status available
<i>prefNetwork</i>	preferred network
<i>currNWInfo</i>	Current Network Info

## 8.993.2 Field Documentation

- 8.993.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`
- 8.993.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`
- 8.993.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`
- 8.993.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`
- 8.993.2.5 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechnology`
- 8.993.2.6 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatAvail`
- 8.993.2.7 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dormancyStatus`
- 8.993.2.8 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipstatAvail`
- 8.993.2.9 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::mipStatus`
- 8.993.2.10 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::netInfoLen`
- 8.993.2.11 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::prefNetwork`
- 8.993.2.12 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::ratMask`
- 8.993.2.13 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_bytes`
- 8.993.2.14 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::rx_pkts`
- 8.993.2.15 `uint32_t unpack_wds_SLQSSetWdsEventCallback_ind_t::soMask`
- 8.993.2.16 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_bytes`
- 8.993.2.17 `uint64_t unpack_wds_SLQSSetWdsEventCallback_ind_t::tx_pkts`
- 8.993.2.18 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::xferStatAvail`

## 8.994 unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t Struct Reference

### Data Fields

- [wdsDhcpv4HwConfig](#) \* [pHwConfig](#)
- [wdsDhcpv4OptionList](#) \* [pRequestOptionList](#)

### 8.994.1 Detailed Description

#### Parameters

<i>pHwConfig</i>	pointer to HW Config structure
<i>pRequestOptionList</i>	pointer to Option List structure to be sent in DHCP request

### 8.994.2 Field Documentation

8.994.2.1 wdsDhcpv4HwConfig\* unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t::pHwConfig

8.994.2.2 wdsDhcpv4OptionList\* unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t::pRequestOptionList

## 8.995 unpack\_wds\_SLQSStartDataSession\_t Struct Reference

### Data Fields

- uint32\_t \* [psid](#)
- uint32\_t \* [pFailureReason](#)
- uint32\_t \* [pVerboseFailReasonType](#)
- uint32\_t \* [pVerboseFailureReason](#)

### 8.995.1 Detailed Description

#### Parameters

<i>psid</i>	<ul style="list-style-type: none"> <li>• Assigned session ID when starting a data session</li> </ul>
<i>pFailureReason</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>pVerboseFail-ReasonType</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>pVerboseFailure-Reason</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>

### 8.995.2 Field Documentation

8.995.2.1 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pFailureReason

8.995.2.2 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::psid

8.995.2.3 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pVerboseFailReasonType

8.995.2.4 uint32\_t\* unpack\_wds\_SLQSStartDataSession\_t::pVerboseFailureReason

## 8.996 unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t Struct Reference

### Data Fields

- uint8\_t [contextId](#)
- uint8\_t [bearerId](#)
- int8\_t [apnName](#) [100]
- uint32\_t [ipv4Address](#)
- uint32\_t [ipv4GWAddress](#)
- uint32\_t [prDNSIPv4Address](#)

- uint32\_t [seDNSIPv4Address](#)
- struct [ipv6AddressInfo](#) [ipv6Address](#)
- struct [ipv6AddressInfo](#) [ipv6GWAddress](#)
- uint16\_t [prDNSIPv6Address](#) [8]
- uint16\_t [seDNSIPv6Address](#) [8]
- uint32\_t [prPCSCFIPv4Address](#)
- uint32\_t [sePCSCFIPv4Address](#)
- uint16\_t [prPCSCFIPv6Address](#) [8]
- uint16\_t [sePCSCFIPv6Address](#) [8]

### 8.996.1 Detailed Description

#### Parameters

<i>contextId</i>	Context Identifier
<i>bearerId</i>	Bearer Identity
<i>apnName</i>	APN name associated with the context id
<i>ipv4Address</i>	IPv4 Address
<i>ipv4GWAddress</i>	IPv4 Gateway Address
<i>prDNSIPv4-Address</i>	Primary DNS IPv4 Address
<i>seDNSIPv4-Address</i>	Secondary DNS IPv4 Address
<i>ipv6Address</i>	IPv6 Address
<i>ipv6GWAddress</i>	IPv6 Gateway Address
<i>prDNSIPv6-Address</i>	Primary IPv6 DNS Address
<i>seDNSIPv6-Address</i>	Secondary IPv6 DNS Address
<i>prPCSCFIPv4-Address</i>	Primary PCSCF IPv4 Address
<i>sePCSCFIPv4-Address</i>	Secondary PCSCF IPv4 Address
<i>prPCSCFIPv6-Address</i>	Primary PCSCF IPv6 Address
<i>sePCSCFIPv6-Address</i>	Secondary PCSCF IPv6 Address

### 8.996.2 Field Documentation

8.996.2.1 int8\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::apnName[100]

8.996.2.2 uint8\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::bearerId

8.996.2.3 uint8\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::contextId

8.996.2.4 uint32\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::ipv4Address

8.996.2.5 uint32\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::ipv4GWAddress

8.996.2.6 struct [ipv6AddressInfo](#) unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::ipv6Address

8.996.2.7 struct [ipv6AddressInfo](#) unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::ipv6GWAddress

8.996.2.8 uint32\_t unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t::prDNSIPv4Address

- 8.996.2.9    `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`
- 8.996.2.10   `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`
- 8.996.2.11   `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`
- 8.996.2.12   `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`
- 8.996.2.13   `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`
- 8.996.2.14   `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`
- 8.996.2.15   `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

## 8.997    UnPackGetProfileSettingOut Struct Reference

### Data Fields

- [UnpackQmiProfileInfo curProfile](#)
- `uint16_t * pExtErrCode`

### 8.997.1    Field Documentation

- 8.997.1.1    `UnpackQmiProfileInfo UnPackGetProfileSettingOut::curProfile`
- 8.997.1.2    `uint16_t* UnPackGetProfileSettingOut::pExtErrCode`

## 8.998    unpackWdsProfileParam Union Reference

### Data Fields

- [LibpackProfile3GPP SlqsProfile3GPP](#)
- [LibpackProfile3GPP2 SlqsProfile3GPP2](#)

### 8.998.1    Field Documentation

- 8.998.1.1    `LibpackProfile3GPP unpackWdsProfileParam::SlqsProfile3GPP`
- 8.998.1.2    `LibpackProfile3GPP2 unpackWdsProfileParam::SlqsProfile3GPP2`

## 8.999    USBCompConfig Struct Reference

### Data Fields

- `BYTE * pUSBComp`

### 8.999.1    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>
---------------------	--

## 8.999.2 Field Documentation

## 8.999.2.1 BYTE\* USBCompConfig::pUSBComp

## 8.1000 USBCompParams Struct Reference

## Data Fields

- [BYTE \\* pUSBComp](#)
- [BYTE \\* pNumSupUSBComps](#)
- [BYTE \\* pSupUSBComps](#)

## 8.1000.1 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(optional 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>
<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>

<p><i>pSupUSB-Comps[OUT]</i></p>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <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>
----------------------------------	--

## 8.1000.2 Field Documentation

8.1000.2.1 **BYTE\*** USBCompParams::pNumSupUSBComps

8.1000.2.2 **BYTE\*** USBCompParams::pSupUSBComps

8.1000.2.3 **BYTE\*** USBCompParams::pUSBComp

## 8.1001 USSDNoWaitIndicationInfo Struct Reference

### Data Fields

- **BYTE \*** [pError](#)
- **BYTE \*** [pFailureCause](#)
- **struct** [USSInfo](#) \* [pUSSDData](#)
- **alphaIDInfo \*** [pAlphaIdentifier](#)

### 8.1001.1 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>

## 8.1001.2 Field Documentation

8.1001.2.1 **alphaIDInfo\*** USSDNoWaitIndicationInfo::pAlphaIdentifier8.1001.2.2 **BYTE\*** USSDNoWaitIndicationInfo::pError8.1001.2.3 **BYTE\*** USSDNoWaitIndicationInfo::pFailureCause8.1001.2.4 **struct USSInfo\*** USSDNoWaitIndicationInfo::pUSSDData

## 8.1002 USSDRespFNetwork Struct Reference

## Data Fields

- char \* [pTypeCode](#)
- char \* [pRespData](#)

## 8.1002.1 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>

## 8.1002.2 Field Documentation

8.1002.2.1 **char\*** USSDRespFNetwork::pRespData8.1002.2.2 **char\*** USSDRespFNetwork::pTypeCode

## 8.1003 USSInfo Struct Reference

## Data Fields

- [BYTE](#) [ussDCS](#)
- [BYTE](#) [ussLen](#)
- [BYTE](#) [ussData](#) [182]

### 8.1003.1 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>

### 8.1003.2 Field Documentation

8.1003.2.1 [BYTE](#) [USSInfo::ussData](#)[182]

8.1003.2.2 [BYTE](#) [USSInfo::ussDCS](#)

8.1003.2.3 [BYTE](#) [USSInfo::ussLen](#)

## 8.1004 USSResp Struct Reference

## Data Fields

- [WORD](#) \* [pfailureCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [struct](#) [USSInfo](#) \* [pUSSDInfo](#)
- [BYTE](#) \* [pCcResultType](#)
- [BYTE](#) \* [pCallId](#)
- [ccSUPSType](#) \* [pCCSuppsType](#)

### 8.1004.1 Field Documentation

8.1004.1.1 [alphaIDInfo](#)\* [USSResp::pAlphaIDInfo](#)

8.1004.1.2 [BYTE](#)\* [USSResp::pCallId](#)

8.1004.1.3 [BYTE](#)\* [USSResp::pCcResultType](#)

8.1004.1.4 [ccSUPSType](#)\* [USSResp::pCCSuppsType](#)

8.1004.1.5 WORD\* USSResp::pfailureCause

8.1004.1.6 struct USSInfo\* USSResp::pUSSDInfo

## 8.1005 UUSInfo Struct Reference

### Data Fields

- [BYTE UUSType](#)
- [BYTE UUSDcs](#)
- [BYTE UUSDatalen](#)
- [BYTE UUSData](#) [255]

### 8.1005.1 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</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• UUS data encoded as per coding scheme</li> </ul>

### 8.1005.2 Field Documentation

8.1005.2.1 BYTE UUSInfo::UUSData[255]

8.1005.2.2 BYTE UUSInfo::UUSDatalen

8.1005.2.3 BYTE UUSInfo::UUSDcs

8.1005.2.4 BYTE UUSInfo::UUSType

## 8.1006 verifyUIMPIN Struct Reference

### Data Fields

- [BYTE pinID](#)
- [BYTE pinLen](#)
- [BYTE pinVal](#) [255]

### 8.1006.1 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>

### 8.1006.2 Field Documentation

8.1006.2.1 BYTE verifyUIMPIN::pinID

8.1006.2.2 BYTE verifyUIMPIN::pinLen

8.1006.2.3 BYTE verifyUIMPIN::pinVal[255]

## 8.1007 voiceALSSelectLineInfo Struct Reference

### Data Fields

- [BYTE lineValue](#)

### 8.1007.1 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>
------------------	--

### 8.1007.2 Field Documentation

#### 8.1007.2.1 BYTE voiceALSSelectLineInfo::lineValue

## 8.1008 voiceALSSetLineSwitchInfo Struct Reference

#### Data Fields

- [BYTE switchOption](#)

### 8.1008.1 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>
---------------------	---

### 8.1008.2 Field Documentation

#### 8.1008.2.1 BYTE voiceALSSetLineSwitchInfo::switchOption

## 8.1009 voiceAnswerCall Struct Reference

#### Data Fields

- [BYTE \\* pCallId](#)

### 8.1009.1 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>
------------------------	--

## 8.1009.2 Field Documentation

8.1009.2.1 **BYTE** \* voiceAnswerCall::pCallId

## 8.1010 voiceBindSubscriptionInfo Struct Reference

### Data Fields

- [BYTE subType](#)

### 8.1010.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

#### Parameters

<i>subType</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>
----------------	--

## 8.1010.2 Field Documentation

8.1010.2.1 **BYTE** voiceBindSubscriptionInfo::subType

## 8.1011 voiceBurstDTMFInfo Struct Reference

### Data Fields

- [burstDTMFInfo BurstDTMFInfo](#)
- [DTMFLengths](#) \* [pBurstDTMFLengths](#)

### 8.1011.1 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>

## 8.1011.2 Field Documentation

8.1011.2.1 **burstDTMFInfo** voiceBurstDTMFInfo::BurstDTMFInfo

8.1011.2.2 DTMFLengths\* voiceBurstDTMFInfo::pBurstDTMFLengths

## 8.1012 voiceCallInfoReq Struct Reference

### Data Fields

- [BYTE](#) callID

#### 8.1012.1 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>
---------------	---

#### 8.1012.2 Field Documentation

8.1012.2.1 [BYTE](#) voiceCallInfoReq::callID

## 8.1013 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](#)

#### 8.1013.1 Detailed Description

This structure contains information of the response parameters associated with a call.

##### Parameters

<i>pCall-Info(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">callInfo</a> for more information.</li></ul>
<i>pRemoteParty-Num(optional)</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(optional)</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.(optional)</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.(optional)</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_SPC_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>pRemoteParty-Name(optional)</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>pUUS-Info(optional)</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>pAlert- Type(optional)</i>	<ul style="list-style-type: none"> <li>Alerting type.</li> <li>Applicable only for 3GPP devices. <ul style="list-style-type: none"> <li>0x00 - ALERTING_LOCAL - Local</li> <li>0x01 - ALERTING_REMOTE - Remote</li> <li>0xFF - Not Available</li> </ul> </li> </ul>
<i>pAlphaID- Info(optional)</i>	<ul style="list-style-type: none"> <li>Applicable only for 3GPP devices.</li> <li>See <a href="#">alphaIDInfo</a> for more information.</li> </ul>
<i>pConnectNum- Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">connectNumInfo</a> for more information.</li> </ul>
<i>pDiag- Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">diagInfo</a> for more information.</li> </ul>
<i>pAlertingPattern</i>	<ul style="list-style-type: none"> <li>Alerting pattern.(optional) <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> <li>0xFF - Not Available</li> </ul> </li> </ul>

## 8.1013.2 Field Documentation

8.1013.2.1 **ULONG\*** voiceCallInfoResp::pAlertingPattern

8.1013.2.2 **BYTE\*** voiceCallInfoResp::pAlertType

8.1013.2.3 **alphaIDInfo\*** voiceCallInfoResp::pAlphaIDInfo

8.1013.2.4 **callInfo\*** voiceCallInfoResp::pCallInfo

8.1013.2.5 **connectNumInfo\*** voiceCallInfoResp::pConnectNumInfo

8.1013.2.6 **diagInfo\*** voiceCallInfoResp::pDiagInfo

8.1013.2.7 **BYTE\*** voiceCallInfoResp::pOTASPStatus

8.1013.2.8 **remotePartyName\*** voiceCallInfoResp::pRemotePartyName

8.1013.2.9 **remotePartyNum\*** voiceCallInfoResp::pRemotePartyNum

8.1013.2.10 **WORD\*** voiceCallInfoResp::pSrvOpt

8.1013.2.11 **UUSInfo\*** voiceCallInfoResp::pUUSInfo

8.1013.2.12 **BYTE\*** voiceCallInfoResp::pVoicePrivacy

## 8.1014 voiceCallRequestParams Struct Reference

### Data Fields

- [BYTE](#) callNumber [81]
- [BYTE \\*](#) pCallType
- [BYTE \\*](#) pCLIRType
- [UUSInfo \\*](#) pUUSInfo
- [CUGInfo \\*](#) pCUGInfo
- [BYTE \\*](#) pEmergencyCategory
- [calledPartySubAdd \\*](#) pCallPartySubAdd
- [ULONG \\*](#) pSvcType

### 8.1014.1 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>pCall-Type(optional)</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>pCLIR-Type(optional)</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(optional)</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(optional)</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(optional)</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(optional)</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>pSvc-Type(optional)</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>

## 8.1014.2 Field Documentation

8.1014.2.1 **BYTE** voiceCallRequestParams::callNumber[81]

8.1014.2.2 **calledPartySubAdd\*** voiceCallRequestParams::pCallPartySubAdd

8.1014.2.3 **BYTE\*** voiceCallRequestParams::pCallType

8.1014.2.4 **BYTE\*** voiceCallRequestParams::pCLIRType

8.1014.2.5 **CUGInfo\*** voiceCallRequestParams::pCUGInfo

8.1014.2.6 **BYTE\*** voiceCallRequestParams::pEmergencyCategory

8.1014.2.7 **ULONG\*** voiceCallRequestParams::pSvcType

8.1014.2.8 **UUSInfo\*** voiceCallRequestParams::pUUSInfo

## 8.1015 voiceCallResponseParams Struct Reference

### Data Fields

- **BYTE \*** pCallID
- **alphaIDInfo \*** pAlphaIDInfo
- **BYTE \*** pCCResultType
- **ccSUPSType \*** pCCSUPSType

### 8.1015.1 Detailed Description

This structure contains Voice Call Response Parameters

#### Parameters

<i>pCallID(optional)</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the dialed call</li> </ul>
<i>pAlphaID-Info(optional)</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>pCCResult-Type(optional)</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>pCCSUPS-Type(optional)</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>

### 8.1015.2 Field Documentation

8.1015.2.1 [alphaIDInfo](#)\* `voiceCallResponseParams::pAlphaIDInfo`

8.1015.2.2 `BYTE`\* `voiceCallResponseParams::pCallID`

8.1015.2.3 `BYTE`\* `voiceCallResponseParams::pCCResultType`

8.1015.2.4 [ccSUPSType](#)\* `voiceCallResponseParams::pCCSUPSType`

## 8.1016 voiceContDTMFinfo Struct Reference

#### Data Fields

- `BYTE` \* [pCallID](#)
- `BYTE` [DTMFdigit](#)

### 8.1016.1 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[IN]</i>	<ul style="list-style-type: none"> <li>• DTMF digit in ASCII.</li> </ul>

## 8.1016.2 Field Documentation

8.1016.2.1 BYTE voiceContDTMFInfo::DTMFdigit

8.1016.2.2 BYTE\* voiceContDTMFInfo::pCallID

## 8.1017 voiceDTMFEventInfo Struct Reference

## Data Fields

- [DTMFInfo DTMFInformation](#)
- [BYTE \\* pOnLength](#)
- [BYTE \\* pOffLength](#)

## 8.1017.1 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

## 8.1017.2 Field Documentation

**8.1017.2.1** DTMFInfo voiceDTMFEventInfo::DTMFInformation**8.1017.2.2** BYTE\* voiceDTMFEventInfo::pOffLength**8.1017.2.3** BYTE\* voiceDTMFEventInfo::pOnLength

## 8.1018 voiceFlashInfo Struct Reference

### Data Fields

- [BYTE](#) \* [pCallID](#)
- [BYTE](#) \* [pFlashPayLd](#)
- [BYTE](#) \* [pFlashType](#)

### 8.1018.1 Detailed Description

This structure contains the flash information associated with a call.

#### Parameters

<i>pCallID</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Unique call identifier associated with the current call.</li> </ul>
<i>pFlashPayLd</i> [I-N](optional)	<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</i> [I-N](optional)	<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>

## 8.1018.2 Field Documentation

**8.1018.2.1** BYTE\* voiceFlashInfo::pCallID**8.1018.2.2** BYTE\* voiceFlashInfo::pFlashPayLd**8.1018.2.3** BYTE\* voiceFlashInfo::pFlashType

## 8.1019 voiceGetAllCallInfo Struct Reference

### Data Fields

- [arrCallInfo](#) \* [pArrCallInfo](#)

- [arrRemotePartyNum](#) \* [pArrRemotePartyNum](#)
- [arrRemotePartyName](#) \* [pArrRemotePartyName](#)
- [arrAlertingType](#) \* [pArrAlertingType](#)
- [arrUUSInfo](#) \* [pArrUUSInfo](#)
- [arrSvcOption](#) \* [pArrSvcOption](#)
- [BYTE](#) \* [pOTASPStatus](#)
- [BYTE](#) \* [pVoicePrivacy](#)
- [arrCallEndReason](#) \* [pArrCallEndReason](#)
- [arrAlphaID](#) \* [pArrAlphaID](#)
- [arrConnectPartyNum](#) \* [pArrConnectPartyNum](#)
- [arrDiagInfo](#) \* [pArrDiagInfo](#)
- [arrCalledPartyNum](#) \* [pArrCalledPartyNum](#)
- [arrRedirPartyNum](#) \* [pArrRedirPartyNum](#)
- [arrAlertingPattern](#) \* [pArrAlertingPattern](#)

### 8.1019.1 Detailed Description

This structure contains information about the response parameters with all the calls originating or terminating from a particular device.

#### Parameters

<i>pArrCall-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCallInfo</a> for more information.</li> </ul>
<i>pArrRemote-Party-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRemotePartyNum</a> for more information.</li> </ul>
<i>pArrRemote-Party-Name(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRemotePartyName</a> for more information.</li> </ul>
<i>pArrAlerting-Type(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlertingType</a> for more information.</li> </ul>
<i>pArrUUS-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrUUSInfo</a> for more information.</li> </ul>
<i>pArrSvc-Option(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrSvcOption</a> for more information.</li> </ul>

<i>pOTASP-Status(optional)</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>pVoice-Privacy(optional)</i>	<ul style="list-style-type: none"> <li>• Voice Privacy.</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>pArrCallEnd-Reason(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCallEndReason</a> for more information.</li> </ul>
<i>pArrAlphaID(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlphaID</a> for more information.</li> </ul>
<i>pArrConnect-Party-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrConnectPartyNum</a> for more information.</li> </ul>
<i>pArrDiag-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrDiagInfo</a> for more information.</li> </ul>
<i>pArrCalledParty-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrCalledPartyNum</a> for more information.</li> </ul>

<i>pArrRedirParty-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrRedirPartyNum</a> for more information.</li> </ul>
<i>pArrAlerting-Pattern(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">arrAlertingPattern</a> for more information.</li> </ul>

## 8.1019.2 Field Documentation

8.1019.2.1 **arrAlertingPattern\*** voiceGetAllCallInfo::pArrAlertingPattern

8.1019.2.2 **arrAlertingType\*** voiceGetAllCallInfo::pArrAlertingType

8.1019.2.3 **arrAlphaID\*** voiceGetAllCallInfo::pArrAlphaID

8.1019.2.4 **arrCalledPartyNum\*** voiceGetAllCallInfo::pArrCalledPartyNum

8.1019.2.5 **arrCallEndReason\*** voiceGetAllCallInfo::pArrCallEndReason

8.1019.2.6 **arrCallInfo\*** voiceGetAllCallInfo::pArrCallInfo

8.1019.2.7 **arrConnectPartyNum\*** voiceGetAllCallInfo::pArrConnectPartyNum

8.1019.2.8 **arrDiagInfo\*** voiceGetAllCallInfo::pArrDiagInfo

8.1019.2.9 **arrRedirPartyNum\*** voiceGetAllCallInfo::pArrRedirPartyNum

8.1019.2.10 **arrRemotePartyName\*** voiceGetAllCallInfo::pArrRemotePartyName

8.1019.2.11 **arrRemotePartyNum\*** voiceGetAllCallInfo::pArrRemotePartyNum

8.1019.2.12 **arrSvcOption\*** voiceGetAllCallInfo::pArrSvcOption

8.1019.2.13 **arrUUSInfo\*** voiceGetAllCallInfo::pArrUUSInfo

8.1019.2.14 **BYTE\*** voiceGetAllCallInfo::pOTASPStatus

8.1019.2.15 **BYTE\*** voiceGetAllCallInfo::pVoicePrivacy

## 8.1020 voiceGetCallBarringReq Struct Reference

### Data Fields

- [BYTE reason](#)
- [BYTE \\* pSvcClass](#)

### 8.1020.1 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_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>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>

## 8.1020.2 Field Documentation

8.1020.2.1 **BYTE**\* voiceGetCallBarringReq::pSvcClass8.1020.2.2 **BYTE** voiceGetCallBarringReq::reason

## 8.1021 voiceGetCallBarringResp Struct Reference

## Data Fields

- **BYTE** \* pSvcClass
- **WORD** \* pFailCause
- **alphaDInfo** \* pAlphaDInfo
- **BYTE** \* pCCResType
- **BYTE** \* pCallID
- **ccSUPSType** \* pCCSUPSType

## 8.1021.1 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.

**8.1021.2 Field Documentation**

8.1021.2.1 **alphaIDInfo\*** voiceGetCallBarringResp::pAlphaIDInfo

8.1021.2.2 **BYTE\*** voiceGetCallBarringResp::pCallID

8.1021.2.3 **BYTE\*** voiceGetCallBarringResp::pCCResType

8.1021.2.4 **ccSUPSType\*** voiceGetCallBarringResp::pCCSUPSType

8.1021.2.5 **WORD\*** voiceGetCallBarringResp::pFailCause

8.1021.2.6 **BYTE\*** voiceGetCallBarringResp::pSvcClass

**8.1022 voiceGetCallFWReq Struct Reference****Data Fields**

- [BYTE Reason](#)
- [BYTE \\* pSvcClass](#)

### 8.1022.1 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>pSvc-Class(optional)</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>

### 8.1022.2 Field Documentation

8.1022.2.1 **BYTE\*** voiceGetCallFWReq::pSvcClass

8.1022.2.2 **BYTE** voiceGetCallFWReq::Reason

## 8.1023 voiceGetCallFWResp Struct Reference

#### Data Fields

- [getCallFWInfo](#) \* [pGetCallFWInfo](#)
- **WORD** \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- **BYTE** \* [pCCResType](#)
- **BYTE** \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)
- [getCallFWExtInfo](#) \* [pGetCallFWExtInfo](#)

### 8.1023.1 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> (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>
<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.

**8.1023.2 Field Documentation**

8.1023.2.1 **alphaIDInfo\*** voiceGetCallFWResp::pAlphaIDInfo

8.1023.2.2 **BYTE\*** voiceGetCallFWResp::pCallID

8.1023.2.3 **BYTE\*** voiceGetCallFWResp::pCCResType

8.1023.2.4 **ccSUPSType\*** voiceGetCallFWResp::pCCSUPSType

8.1023.2.5 **WORD\*** voiceGetCallFWResp::pFailCause

8.1023.2.6 **getCallFWExtInfo\*** voiceGetCallFWResp::pGetCallFWExtInfo

8.1023.2.7 **getCallFWInfo\*** voiceGetCallFWResp::pGetCallFWInfo

**8.1024 voiceGetCallWaitInfo Struct Reference**

## Data Fields

- [BYTE](#) \* [pSvcClass](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.1024.1 Detailed Description

This structure contains Voice Get Call Waiting 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> [OUT]	<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> [OUT]	<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> [OUT]	<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> [OUT]	<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> [OUT]	<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.

### 8.1024.2 Field Documentation

- 8.1024.2.1 **alphaIDInfo\*** voiceGetCallWaitInfo::pAlphaIDInfo
- 8.1024.2.2 **BYTE\*** voiceGetCallWaitInfo::pCallID
- 8.1024.2.3 **BYTE\*** voiceGetCallWaitInfo::pCCResType
- 8.1024.2.4 **ccSUPSType\*** voiceGetCallWaitInfo::pCCSUPSType
- 8.1024.2.5 **WORD\*** voiceGetCallWaitInfo::pFailCause
- 8.1024.2.6 **BYTE\*** voiceGetCallWaitInfo::pSvcClass

## 8.1025 voiceGetCLIPResp Struct Reference

### Data Fields

- [CLIPResp](#) \* [pCLIPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.1025.1 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.

**8.1025.2 Field Documentation**

8.1025.2.1 **alphaIDInfo\*** voiceGetCLIPResp::pAlphaIDInfo

8.1025.2.2 **BYTE\*** voiceGetCLIPResp::pCallID

8.1025.2.3 **BYTE\*** voiceGetCLIPResp::pCCResType

8.1025.2.4 **ccSUPSType\*** voiceGetCLIPResp::pCCSUPSType

8.1025.2.5 **CLIPResp\*** voiceGetCLIPResp::pCLIPResp

8.1025.2.6 **WORD\*** voiceGetCLIPResp::pFailCause

**8.1026 voiceGetCLIRResp Struct Reference****Data Fields**

- [CLIRResp](#) \* [pCLIRResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

**8.1026.1 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.

**8.1026.2 Field Documentation**

8.1026.2.1 **alphaIDInfo\*** voiceGetCLIRResp::pAlphaIDInfo

8.1026.2.2 **BYTE\*** voiceGetCLIRResp::pCallID

8.1026.2.3 **BYTE\*** voiceGetCLIRResp::pCCResType

8.1026.2.4 **ccSUPSType\*** voiceGetCLIRResp::pCCSUPSType

8.1026.2.5 **CLIRResp\*** voiceGetCLIRResp::pCLIRResp

8.1026.2.6 **WORD\*** voiceGetCLIRResp::pFailCause

**8.1027 voiceGetCNAPResp Struct Reference****Data Fields**

- [CNAPResp](#) \* [pCNAPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

**8.1027.1 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 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.

## 8.1027.2 Field Documentation

8.1027.2.1 **alphaIDInfo\*** voiceGetCNAPResp::pAlphaIDInfo

8.1027.2.2 **BYTE\*** voiceGetCNAPResp::pCallID

8.1027.2.3 **BYTE\*** voiceGetCNAPResp::pCCResType

8.1027.2.4 **ccSUPSType\*** voiceGetCNAPResp::pCCSUPSType

8.1027.2.5 **CNAPResp\*** voiceGetCNAPResp::pCNAPResp

8.1027.2.6 **WORD\*** voiceGetCNAPResp::pFailCause

## 8.1028 voiceGetCOLPResp Struct Reference

## Data Fields

- [COLPResp](#) \* [pCOLPResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

## 8.1028.1 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.

## 8.1028.2 Field Documentation

8.1028.2.1 [alphaIDInfo](#)\* [voiceGetCOLPResp::pAlphaIDInfo](#)

8.1028.2.2 **BYTE\*** voiceGetCOLPResp::pCallID

8.1028.2.3 **BYTE\*** voiceGetCOLPResp::pCCResType

8.1028.2.4 **ccSUPSType\*** voiceGetCOLPResp::pCCSUPSType

8.1028.2.5 **COLPResp\*** voiceGetCOLPResp::pCOLPResp

8.1028.2.6 **WORD\*** voiceGetCOLPResp::pFailCause

## 8.1029 voiceGetCOLRResp Struct Reference

### Data Fields

- [COLRResp](#) \* [pCOLRResp](#)
- [WORD](#) \* [pFailCause](#)
- [alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [BYTE](#) \* [pCCResType](#)
- [BYTE](#) \* [pCallID](#)
- [ccSUPSType](#) \* [pCCSUPSType](#)

### 8.1029.1 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.

**8.1029.2 Field Documentation**

8.1029.2.1 **alphaIDInfo\*** voiceGetCOLRResp::pAlphaIDInfo

8.1029.2.2 **BYTE\*** voiceGetCOLRResp::pCallID

8.1029.2.3 **BYTE\*** voiceGetCOLRResp::pCCResType

8.1029.2.4 **ccSUPSType\*** voiceGetCOLRResp::pCCSUPSType

8.1029.2.5 **COLRResp\*** voiceGetCOLRResp::pCOLRResp

8.1029.2.6 **WORD\*** voiceGetCOLRResp::pFailCause

**8.1030 voiceGetConfigReq Struct Reference****Data Fields**

- **BYTE \*** [pAutoAnswer](#)
- **BYTE \*** [pAirTimer](#)
- **BYTE \*** [pRoamTimer](#)
- **BYTE \*** [pTTYMode](#)
- **BYTE \*** [pPrefVoiceSO](#)
- **BYTE \*** [pAMRStatus](#)
- **BYTE \*** [pPrefVoicePrivacy](#)
- **BYTE \*** [pNameID](#)
- **BYTE \*** [pVoiceDomainPref](#)

**8.1030.1 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.

### 8.1030.2 Field Documentation

8.1030.2.1 **BYTE\*** voiceGetConfigReq::pAirTimer

8.1030.2.2 **BYTE\*** voiceGetConfigReq::pAMRStatus

8.1030.2.3 **BYTE\*** voiceGetConfigReq::pAutoAnswer

8.1030.2.4 **BYTE\*** voiceGetConfigReq::pNamID

8.1030.2.5 **BYTE\*** voiceGetConfigReq::pPrefVoicePrivacy

8.1030.2.6 **BYTE\*** voiceGetConfigReq::pPrefVoiceSO

8.1030.2.7 **BYTE\*** voiceGetConfigReq::pRoamTimer

8.1030.2.8 **BYTE\*** voiceGetConfigReq::pTTYMode

8.1030.2.9 **BYTE\*** voiceGetConfigReq::pVoiceDomainPref

## 8.1031 voiceGetConfigResp Struct Reference

### Data Fields

- **BYTE \*** pAutoAnswerStat
- **airTimer \*** pAirTimerCnt
- **roamTimer \*** pRoamTimerCnt
- **BYTE \*** pCurrTTYMode
- **prefVoiceSO \*** pCurPrefVoiceSO
- **curAMRConfig \*** pCurAMRConfig
- **BYTE \*** pCurVoicePrivacyPref
- **BYTE \*** pCurVoiceDomainPref

### 8.1031.1 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>pCurAMRConfig(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_AMR_CODEC_PREFERENCE_CONFIG_I.</li> <li>• See <a href="#">curAMRConfig</a> for more information</li> </ul>
<i>pCurVoicePrivacyPref(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>pCurVoiceDomainPref(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.

**8.1031.2 Field Documentation**

8.1031.2.1 **airTimer\*** voiceGetConfigResp::pAirTimerCnt

8.1031.2.2 **BYTE\*** voiceGetConfigResp::pAutoAnswerStat

8.1031.2.3 **curAMRConfig\*** voiceGetConfigResp::pCurAMRConfig

8.1031.2.4 **prefVoiceSO\*** voiceGetConfigResp::pCurPrefVoiceSO

8.1031.2.5 **BYTE\*** voiceGetConfigResp::pCurrTTYMode

8.1031.2.6 **BYTE\*** voiceGetConfigResp::pCurVoiceDomainPref

8.1031.2.7 **BYTE\*** voiceGetConfigResp::pCurVoicePrivacyPref

8.1031.2.8 **roamTimer\*** voiceGetConfigResp::pRoamTimerCnt

**8.1032 voiceIndicationRegisterInfo Struct Reference**

## Data Fields

- [BYTE \\* pRegDTMFEvents](#)
- [BYTE \\* pRegVoicePrivacyEvents](#)
- [BYTE \\* pSuppsNotifEvents](#)

### 8.1032.1 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.

### 8.1032.2 Field Documentation

8.1032.2.1 **BYTE\*** voicelIndicationRegisterInfo::pRegDTMFEvents

8.1032.2.2 **BYTE\*** voicelIndicationRegisterInfo::pRegVoicePrivacyEvents

8.1032.2.3 **BYTE\*** voicelIndicationRegisterInfo::pSuppsNotifEvents

## 8.1033 voicelInfoRec Struct Reference

## Data Fields

- [BYTE callID](#)
- [signalInfo \\* pSignalInfo](#)
- [callerIDInfo \\* pCallerIDInfo](#)

- [BYTE \\* pDisplInfo](#)
- [BYTE \\* pExtDisplInfo](#)
- [BYTE \\* pCallerNameInfo](#)
- [BYTE \\* pCallWaitInd](#)
- [connectNumInfo \\* pConnectNumInfo](#)
- [connectNumInfo \\* pCallingPartyInfo](#)
- [calledPartyInfo \\* pCalledPartyInfo](#)
- [redirNumInfo \\* pRedirNumInfo](#)
- [BYTE \\* pCLIRCause](#)
- [NSSAudioCtrl \\* pNSSAudioCtrl](#)
- [BYTE \\* pNSSRelease](#)
- [lineCtrlInfo \\* pLineCtrlInfo](#)
- [extDispRecInfo \\* pExtDispRecInfo](#)

### 8.1033.1 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>pDisplInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Display Information</li> </ul>
<i>pExtDisplInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Extended Display Information</li> </ul>
<i>pCallerName- Info</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>pConnectNum- Info</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>pCallingParty- Info</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>pCalledParty- Info</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>

## 8.1033.2 Field Documentation

8.1033.2.1 **BYTE** voicelInfoRec::callID

8.1033.2.2 **calledPartyInfo\*** voicelInfoRec::pCalledPartyInfo

8.1033.2.3 **callerIDInfo\*** voicelInfoRec::pCallerIDInfo

8.1033.2.4 **BYTE\*** voicelInfoRec::pCallerNameInfo

8.1033.2.5 **connectNumInfo\*** voicelInfoRec::pCallingPartyInfo

8.1033.2.6 **BYTE\*** voicelInfoRec::pCallWaitInd

8.1033.2.7 **BYTE\*** voicelInfoRec::pCLIRCause

8.1033.2.8 **connectNumInfo\*** voicelInfoRec::pConnectNumInfo

8.1033.2.9 **BYTE\*** voicelInfoRec::pDispInfo

8.1033.2.10 **BYTE\*** voicelInfoRec::pExtDispInfo

8.1033.2.11 **extDispRecInfo\*** voicelInfoRec::pExtDispRecInfo

8.1033.2.12 **lineCtrlInfo\*** voicelInfoRec::pLineCtrlInfo

8.1033.2.13 **NSSAudioCtrl\*** voicelInfoRec::pNSSAudioCtrl

8.1033.2.14 **BYTE\*** voicelInfoRec::pNSSRelease

8.1033.2.15 **redirNumInfo\*** `voiceInfoRec::pRedirNumInfo`

8.1033.2.16 **signalInfo\*** `voiceInfoRec::pSignalInfo`

## 8.1034 **voiceManageCallsReq** Struct Reference

### Data Fields

- [BYTE SUPSType](#)
- [BYTE \\* pCallID](#)

### 8.1034.1 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_CONFERECE_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>

### 8.1034.2 Field Documentation

8.1034.2.1 **BYTE\*** `voiceManageCallsReq::pCallID`

8.1034.2.2 **BYTE** `voiceManageCallsReq::SUPSType`

## 8.1035 **voiceManageCallsResp** Struct Reference

## Data Fields

- [WORD](#) \* [pFailCause](#)

### 8.1035.1 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>
-------------------	---

### 8.1035.2 Field Documentation

8.1035.2.1 [WORD](#)\* [voiceManageCallsResp::pFailCause](#)

## 8.1036 voiceOrigUSSDNoWaitInfo Struct Reference

## Data Fields

- struct [USSInfo](#) [USSInformation](#)

### 8.1036.1 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>
-----------------------	---

### 8.1036.2 Field Documentation

8.1036.2.1 struct [USSInfo](#) [voiceOrigUSSDNoWaitInfo::USSInformation](#)

## 8.1037 voiceOTASPStatusInfo Struct Reference

## Data Fields

- [BYTE](#) [callID](#)
- [BYTE](#) [OTASPStatus](#)

### 8.1037.1 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_SPC_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>

## 8.1037.2 Field Documentation

8.1037.2.1 BYTE voiceOTASPStatusInfo::callID

8.1037.2.2 BYTE voiceOTASPStatusInfo::OTASPStatus

## 8.1038 voicePrivacyInfo Struct Reference

## Data Fields

- [BYTE callID](#)
- [BYTE voicePrivacy](#)

## 8.1038.1 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

## 8.1038.2 Field Documentation

8.1038.2.1 BYTE voicePrivacyInfo::callID

8.1038.2.2 BYTE voicePrivacyInfo::voicePrivacy

## 8.1039 voiceSetAllCallStatusCbkJInfo Struct Reference

## Data Fields

- [arrCallInfo](#) [arrCallInformation](#)
- [arrRemotePartyNum](#) \* [pArrRemotePartyNum](#)
- [arrRemotePartyName](#) \* [pArrRemotePartyName](#)
- [arrAlertingType](#) \* [pArrAlertingType](#)
- [arrSvcOption](#) \* [pArrSvcOption](#)
- [arrCallEndReason](#) \* [pArrCallEndReason](#)
- [arrAlphaID](#) \* [pArrAlphaID](#)
- [arrConnectPartyNum](#) \* [pArrConnectPartyNum](#)
- [arrDiagInfo](#) \* [pArrDiagInfo](#)
- [arrCalledPartyNum](#) \* [pArrCalledPartyNum](#)
- [arrRedirPartyNum](#) \* [pArrRedirPartyNum](#)
- [arrAlertingPattern](#) \* [pArrAlertingPattern](#)

## 8.1039.1 Detailed Description

This structure contains VoiceCall Information parameters. [arrCallInformation](#) will be populated in case of change in the call information. Other parameters are optional therefore are populated based on device and technology type being used.

## Parameters

<i>arrCallInformation</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>pArrRemotePartyName</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>pArrCallEndReason</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. <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrConnectPartyNum</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. <ul style="list-style-type: none"> <li>– Applicable for both "3GPP/3GPP2"</li> </ul> </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. <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrCalledPartyNum</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. <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrRedirPartyNum</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. <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>
<i>pArrAlertingPattern</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. <ul style="list-style-type: none"> <li>– Applicable only for "3GPP"</li> </ul> </li> </ul>

## Note

Optional paramters would be NULL, if not received from the device.

## 8.1039.2 Field Documentation

- 8.1039.2.1 `arrCallInfo` `voiceSetAllCallStatusCbkiInfo::arrCallInfomation`
- 8.1039.2.2 `arrAlertingPattern*` `voiceSetAllCallStatusCbkiInfo::pArrAlertingPattern`
- 8.1039.2.3 `arrAlertingType*` `voiceSetAllCallStatusCbkiInfo::pArrAlertingType`
- 8.1039.2.4 `arrAlphaID*` `voiceSetAllCallStatusCbkiInfo::pArrAlphaID`
- 8.1039.2.5 `arrCalledPartyNum*` `voiceSetAllCallStatusCbkiInfo::pArrCalledPartyNum`
- 8.1039.2.6 `arrCallEndReason*` `voiceSetAllCallStatusCbkiInfo::pArrCallEndReason`
- 8.1039.2.7 `arrConnectPartyNum*` `voiceSetAllCallStatusCbkiInfo::pArrConnectPartyNum`
- 8.1039.2.8 `arrDiagInfo*` `voiceSetAllCallStatusCbkiInfo::pArrDiagInfo`
- 8.1039.2.9 `arrRedirPartyNum*` `voiceSetAllCallStatusCbkiInfo::pArrRedirPartyNum`
- 8.1039.2.10 `arrRemotePartyName*` `voiceSetAllCallStatusCbkiInfo::pArrRemotePartyName`
- 8.1039.2.11 `arrRemotePartyNum*` `voiceSetAllCallStatusCbkiInfo::pArrRemotePartyNum`
- 8.1039.2.12 `arrSvcOption*` `voiceSetAllCallStatusCbkiInfo::pArrSvcOption`

## 8.1040 voiceSetCallBarringPwdInfo Struct Reference

### Data Fields

- [BYTE Reason](#)
- [BYTE oldPasswd](#) [4]
- [BYTE newPasswd](#) [4]
- [BYTE newPasswdAgain](#) [4]

### 8.1040.1 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_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>oldPasswd[PASSWORD_LENGTH]</i>	<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[PASSWORD_LENGTH]</i>	<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[PASSWORD_LENGTH]</i>	<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>

## 8.1040.2 Field Documentation

8.1040.2.1 BYTE voiceSetCallBarringPwdInfo::newPasswd[4]

8.1040.2.2 BYTE voiceSetCallBarringPwdInfo::newPasswdAgain[4]

8.1040.2.3 BYTE voiceSetCallBarringPwdInfo::oldPasswd[4]

8.1040.2.4 BYTE voiceSetCallBarringPwdInfo::Reason

## 8.1041 voiceSetCallBarringPwdResp Struct Reference

## Data Fields

- WORD \* pFailCause
- alphaIDInfo \* pAlphaIDInfo
- BYTE \* pCCResType
- BYTE \* pCallID
- ccSUPSType \* pCCSUPSType

### 8.1041.1 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.

### 8.1041.2 Field Documentation

8.1041.2.1 **alphaIDInfo\*** voiceSetCallBarringPwdResp::pAlphaIDInfo

8.1041.2.2 **BYTE\*** voiceSetCallBarringPwdResp::pCallID

8.1041.2.3 **BYTE\*** voiceSetCallBarringPwdResp::pCCResType

8.1041.2.4 **ccSUPSType\*** voiceSetCallBarringPwdResp::pCCSUPSType

8.1041.2.5 **WORD\*** voiceSetCallBarringPwdResp::pFailCause

## 8.1042 voiceSetConfigReq Struct Reference

#### Data Fields

- **BYTE \*** [pAutoAnswer](#)

- [airTimer](#) \* [pAirTimerConfig](#)
- [roamTimer](#) \* [pRoamTimerConfig](#)
- [BYTE](#) \* [pTTYMode](#)
- [prefVoiceSO](#) \* [pPrefVoiceSO](#)
- [BYTE](#) \* [pPrefVoiceDomain](#)

### 8.1042.1 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>pRoamTimer-Config</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.

### 8.1042.2 Field Documentation

- 8.1042.2.1 **airTimer\*** voiceSetConfigReq::pAirTimerConfig
- 8.1042.2.2 **BYTE\*** voiceSetConfigReq::pAutoAnswer
- 8.1042.2.3 **BYTE\*** voiceSetConfigReq::pPrefVoiceDomain
- 8.1042.2.4 **prefVoiceSO\*** voiceSetConfigReq::pPrefVoiceSO
- 8.1042.2.5 **roamTimer\*** voiceSetConfigReq::pRoamTimerConfig
- 8.1042.2.6 **BYTE\*** voiceSetConfigReq::pTTYMode

## 8.1043 voiceSetConfigResp Struct Reference

### Data Fields

- **BYTE \*** pAutoAnsStatus
- **BYTE \*** pAirTimerStatus
- **BYTE \*** pRoamTimerStatus
- **BYTE \*** pTTYConfigStatus
- **BYTE \*** pPrefVoiceSOStatus
- **BYTE \*** pVoiceDomainPrefStatus

### 8.1043.1 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.

**8.1043.2 Field Documentation**

8.1043.2.1 **BYTE\*** voiceSetConfigResp::pAirTimerStatus

8.1043.2.2 **BYTE\*** voiceSetConfigResp::pAutoAnsStatus

8.1043.2.3 **BYTE\*** voiceSetConfigResp::pPrefVoiceSOStatus

8.1043.2.4 **BYTE\*** voiceSetConfigResp::pRoamTimerStatus

8.1043.2.5 **BYTE\*** voiceSetConfigResp::pTTYConfigStatus

8.1043.2.6 **BYTE\*** voiceSetConfigResp::pVoiceDomainPrefStatus

**8.1044 voiceSetPrefPrivacy Struct Reference****Data Fields**

- [BYTE](#) privacyPref

**8.1044.1 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>
--------------------	---

## 8.1044.2 Field Documentation

8.1044.2.1 BYTE voiceSetPrefPrivacy::privacyPref

## 8.1045 voiceSetSUPSServiceReq Struct Reference

## Data Fields

- [BYTE voiceSvc](#)
- [BYTE reason](#)
- [BYTE \\* pServiceClass](#)
- [BYTE \\* pCallBarringPasswd](#)
- [BYTE \\* pCallForwardingNumber](#)
- [BYTE \\* pTimerVal](#)
- [callFwdTypeAndPlan \\* pCallFwdTypeAndPlan](#)

## 8.1045.1 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_OUTGOINGINTEXTOTHOME 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>

## 8.1045.2 Field Documentation

8.1045.2.1 **BYTE\*** voiceSetSUPSServiceReq::pCallBarringPasswd

8.1045.2.2 **BYTE\*** voiceSetSUPSServiceReq::pCallForwardingNumber

8.1045.2.3 **callFwdTypeAndPlan\*** voiceSetSUPSServiceReq::pCallFwdTypeAndPlan

8.1045.2.4 **BYTE\*** voiceSetSUPSServiceReq::pServiceClass

8.1045.2.5 **BYTE\*** voiceSetSUPSServiceReq::pTimerVal

8.1045.2.6 **BYTE** voiceSetSUPSServiceReq::reason

8.1045.2.7 **BYTE** voiceSetSUPSServiceReq::voiceSvc

## 8.1046 voiceSetSUPSServiceResp Struct Reference

### Data Fields

- **WORD\*** [pFailCause](#)
- **alphaIDInfo\*** [pAlphaIDInfo](#)
- **BYTE\*** [pCCResultType](#)
- **BYTE\*** [pCallID](#)
- **ccSUPSType\*** [pCCSUPSType](#)

### 8.1046.1 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>

## 8.1046.2 Field Documentation

8.1046.2.1 **alphaIDInfo\*** voiceSetSUPSServiceResp::pAlphaIDInfo

8.1046.2.2 **BYTE\*** voiceSetSUPSServiceResp::pCallID

8.1046.2.3 **BYTE\*** voiceSetSUPSServiceResp::pCCResultType

8.1046.2.4 **ccSUPSType\*** voiceSetSUPSServiceResp::pCCSUPSType

8.1046.2.5 **WORD\*** voiceSetSUPSServiceResp::pFailCause

## 8.1047 voiceStopContDTMFInfo Struct Reference

### Data Fields

- [BYTE callID](#)

### 8.1047.1 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>
-------------------------	--

## 8.1047.2 Field Documentation

## 8.1047.2.1 BYTE voiceStopContDTMFInfo::callID

## 8.1048 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

## 8.1048.1 Detailed Description

This structure contains the parameters passed for SLQSVoiceSetSUPSCallBack by the device.

## Parameters

<i>SUPS-Information</i> (mandatory)	See <a href="#">SUPSInfo</a> for more information.
<i>pSvc-Class</i> (optional)	<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</i> (optional)	<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

## 8.1048.2 Field Documentation

- 8.1048.2.1 **alphaIDInfo\*** voiceSUPSInfo::pAlphaIDInfo
- 8.1048.2.2 **BYTE\*** voiceSUPSInfo::pCallBarPasswd
- 8.1048.2.3 **getCallFWInfo\*** voiceSUPSInfo::pCallFwdInfo
- 8.1048.2.4 **BYTE\*** voiceSUPSInfo::pCallFWNum
- 8.1048.2.5 **BYTE\*** voiceSUPSInfo::pCallFWTimerVal
- 8.1048.2.6 **BYTE\*** voiceSUPSInfo::pCallIID
- 8.1048.2.7 **CLIPResp\*** voiceSUPSInfo::pCLIPstatus
- 8.1048.2.8 **CLIRResp\*** voiceSUPSInfo::pCLIRstatus
- 8.1048.2.9 **CNAPResp\*** voiceSUPSInfo::pCNAPstatus
- 8.1048.2.10 **COLPResp\*** voiceSUPSInfo::pCOLPstatus
- 8.1048.2.11 **COLRResp\*** voiceSUPSInfo::pCOLRstatus
- 8.1048.2.12 **BYTE\*** voiceSUPSInfo::pDataSrc
- 8.1048.2.13 **WORD\*** voiceSUPSInfo::pFailCause
- 8.1048.2.14 **newPwdData\*** voiceSUPSInfo::pNewPwdData
- 8.1048.2.15 **BYTE\*** voiceSUPSInfo::pReason
- 8.1048.2.16 **BYTE\*** voiceSUPSInfo::pSvcClass
- 8.1048.2.17 **struct USSInfo\*** voiceSUPSInfo::pUSSInfo
- 8.1048.2.18 **SUPSInfo** voiceSUPSInfo::SUPSInformation

## 8.1049 voiceSUPSNotification Struct Reference

## Data Fields

- [BYTE](#) callID
- [BYTE](#) notifType
- [WORD](#) \* [pCUGIndex](#)
- [ECTNum](#) \* [pECTNum](#)

### 8.1049.1 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) Refer <a href="#">ECTNum</a> for details.</li> </ul>

## Note

None

## 8.1049.2 Field Documentation

8.1049.2.1 BYTE voiceSUPSNotification::callID

8.1049.2.2 BYTE voiceSUPSNotification::notifType

8.1049.2.3 WORD\* voiceSUPSNotification::pCUGIndex

8.1049.2.4 ECTNum\* voiceSUPSNotification::pECTNum

## 8.1050 wcdmaCellInfo Struct Reference

## Data Fields

- WORD psc
- SHORT cpich\_rscp
- SHORT cpich\_ecno
- SHORT srxlev

## 8.1050.1 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>

## 8.1050.2 Field Documentation

8.1050.2.1 SHORT wcdmaCellInfo::cpich\_ecno

8.1050.2.2 SHORT wcdmaCellInfo::cpich\_rscp

8.1050.2.3 **WORD** wcdmaCellInfo::psc

8.1050.2.4 **SHORT** wcdmaCellInfo::srxlev

## 8.1051 WCDMAECIOThresh Struct Reference

### Data Fields

- [BYTE](#) WCDMAECIOThreshListLen
- [WORD](#) \* pWCDMAECIOThreshList

### 8.1051.1 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>

### 8.1051.2 Field Documentation

8.1051.2.1 **WORD\*** WCDMAECIOThresh::pWCDMAECIOThreshList

8.1051.2.2 **BYTE** WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.1052 WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- [ULONG](#) wcdmaRRCTest
- [BYTE](#) umtsLTENbrCellLen
- [umtsLTENbrCell](#) UMTSLTENbrCell [255]

### 8.1052.1 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 * WCDMA RRC State is IDLE</li> <li>– 0x01 - NAS_WCDMA_RRC_STATE_CELL_PCH * WCDMA RRC state is CELL_PCH</li> <li>– 0x02 - NAS_WCDMA_RRC_STATE_URA_PCH * WCDMA RRC state is URA_PCH</li> <li>– 0x03 - NAS_WCDMA_RRC_STATE_CELL_FACH * WCDMA RRC state is CELL_FACH</li> <li>– 0x04 - NAS_WCDMA_RRC_STATE_CELL_DCH * WCDMA RRC state is CELL_DCH</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>

## 8.1052.2 Field Documentation

8.1052.2.1 `umtsLTENbrCell` WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]8.1052.2.2 `BYTE` WCDMAInfoLTENeighborCell::umtsLTENbrCellLen8.1052.2.3 `ULONG` WCDMAInfoLTENeighborCell::wcdmaRRCState

## 8.1053 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`

## 8.1053.1 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>pReferenceNum</i> [OUT]	<ul style="list-style-type: none"> <li>• Reference number of the sms</li> </ul>
<i>pTotalNum</i> [OUT]	<ul style="list-style-type: none"> <li>• Total number of the concatenated message</li> </ul>
<i>pPartNum</i> [OUT]	<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</li> <li>• concatenated SMS.</li> </ul>

## 8.1053.2 Field Documentation

8.1053.2.1 BYTE wcdmaLongMsgDecodingParams::Date[0x09]

8.1053.2.2 BOOL\* wcdmaLongMsgDecodingParams::plsUDHPresent

- 8.1053.2.3 **BYTE\*** wcdmaLongMsgDecodingParams::pMessage
- 8.1053.2.4 **BYTE\*** wcdmaLongMsgDecodingParams::pPartNum
- 8.1053.2.5 **BYTE\*** wcdmaLongMsgDecodingParams::pReferenceNum
- 8.1053.2.6 **CHAR\*** wcdmaLongMsgDecodingParams::pScAddr
- 8.1053.2.7 **BYTE\*** wcdmaLongMsgDecodingParams::pScAddrLength
- 8.1053.2.8 **CHAR\*** wcdmaLongMsgDecodingParams::pSenderAddr
- 8.1053.2.9 **BYTE\*** wcdmaLongMsgDecodingParams::pSenderAddrLength
- 8.1053.2.10 **CHAR\*** wcdmaLongMsgDecodingParams::pTextMsg
- 8.1053.2.11 **BYTE\*** wcdmaLongMsgDecodingParams::pTextMsgLength
- 8.1053.2.12 **BYTE\*** wcdmaLongMsgDecodingParams::pTotalNum
- 8.1053.2.13 **BYTE** wcdmaLongMsgDecodingParams::Time[0x09]

## 8.1054 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]

### 8.1054.1 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 filed 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>

## 8.1054.2 Field Documentation

8.1054.2.1 **BYTE** wcdmaMsgDecodingParams::Date[0x09]

8.1054.2.2 **BYTE\*** wcdmaMsgDecodingParams::pMessage

8.1054.2.3 **CHAR\*** wcdmaMsgDecodingParams::pScAddr

8.1054.2.4 **BYTE\*** wcdmaMsgDecodingParams::pScAddrLength

8.1054.2.5 **CHAR\*** wcdmaMsgDecodingParams::pSenderAddr

8.1054.2.6 **BYTE\*** wcdmaMsgDecodingParams::pSenderAddrLength

8.1054.2.7 **CHAR\*** wcdmaMsgDecodingParams::pTextMsg

8.1054.2.8 **BYTE\*** wcdmaMsgDecodingParams::pTextMsgLength

8.1054.2.9 **BYTE** wcdmaMsgDecodingParams::Time[0x09]

## 8.1055 wcdmaMsgEncodingParams Struct Reference

### Data Fields

- [ULONG](#) messageSize
- [CHAR \\*](#) pDestAddr
- [CHAR \\*](#) pTextMsg
- [CHAR \\*](#) pPDUMessage
- [BYTE](#) alphabet

### 8.1055.1 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>

### 8.1055.2 Field Documentation

8.1055.2.1 **BYTE** wcdmaMsgEncodingParams::alphabet

8.1055.2.2 **ULONG** wcdmaMsgEncodingParams::messageSize

8.1055.2.3 **CHAR\*** wcdmaMsgEncodingParams::pDestAddr

8.1055.2.4 **CHAR\*** wcdmaMsgEncodingParams::pPDUMessage

8.1055.2.5 **CHAR\*** wcdmaMsgEncodingParams::pTextMsg

## 8.1056 WCDMARSSIThresh Struct Reference

### Data Fields

- [BYTE WCDMARSSIThreshListLen](#)
- [WORD \\* pWCDMARSSIThreshList](#)

### 8.1056.1 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>
-----------------------------------	--

## 8.1056.2 Field Documentation

8.1056.2.1 **WORD\*** WCDMARSSIThresh::pWCDMARSSIThreshList

8.1056.2.2 **BYTE** WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.1057 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

### 8.1057.1 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>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>
<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> <li>• In case of two-digit MNC values, the third (unused) digit is set to 0xFF. For example, 15 (a two-digit MNC) is reported using the byte stream 0x31 0x35 0xFF.</li> </ul>

<i>hsCallStatus-Valid</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>

## 8.1057.2 Field Documentation

- 8.1057.2.1 **ULONG** WCDMASysInfo::cellId
- 8.1057.2.2 **BYTE** WCDMASysInfo::cellIdValid
- 8.1057.2.3 **BYTE** WCDMASysInfo::hsCallStatus
- 8.1057.2.4 **BYTE** WCDMASysInfo::hsCallStatusValid
- 8.1057.2.5 **BYTE** WCDMASysInfo::hsInd
- 8.1057.2.6 **BYTE** WCDMASysInfo::hsIndValid
- 8.1057.2.7 **WORD** WCDMASysInfo::lac
- 8.1057.2.8 **BYTE** WCDMASysInfo::lacValid
- 8.1057.2.9 **BYTE** WCDMASysInfo::MCC[3]
- 8.1057.2.10 **BYTE** WCDMASysInfo::MNC[3]
- 8.1057.2.11 **BYTE** WCDMASysInfo::networkIdValid
- 8.1057.2.12 **WORD** WCDMASysInfo::psc
- 8.1057.2.13 **BYTE** WCDMASysInfo::pscValid
- 8.1057.2.14 **BYTE** WCDMASysInfo::regRejectInfoValid
- 8.1057.2.15 **BYTE** WCDMASysInfo::rejCause
- 8.1057.2.16 **BYTE** WCDMASysInfo::rejectSrvDomain
- 8.1057.2.17 **sysInfoCommon** WCDMASysInfo::sysInfoWCDMA

## 8.1058 wcdmaUARFCN Struct Reference

### Data Fields

- [BYTE](#) *status*
- [ULONG](#) *uarfcn*

### 8.1058.1 Detailed Description

This structure contains the parameters for WCDMA UARFCN.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• 0 - Disable</li> <li>• 1 - Enable</li> </ul>
<i>uarfcn</i>	<ul style="list-style-type: none"> <li>• UARFCN to which UE is locked</li> </ul>

## 8.1058.2 Field Documentation

8.1058.2.1 **BYTE** wcdmaUARFCN::status

8.1058.2.2 **ULONG** wcdmaUARFCN::uarfcn

## 8.1059 wds\_currNetworkInfo Struct Reference

### Data Fields

- uint8\_t [NetworkType](#)
- uint32\_t [RATMask](#)
- uint32\_t [SOMask](#)

### 8.1059.1 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> <li>• CDMA EV-DO Rev B 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> <li>– 0x10 - MMPA</li> <li>– 0x20 - MMPA_EHRPD</li> </ul> </li> </ul>
---------------	--

## 8.1059.2 Field Documentation

8.1059.2.1 `uint8_t wds_currNetworkInfo::NetworkType`

8.1059.2.2 `uint32_t wds_currNetworkInfo::RATMask`

8.1059.2.3 `uint32_t wds_currNetworkInfo::SOMask`

## 8.1060 wds\_Domain Struct Reference

### Data Fields

- `uint16_t domainLen`
- `uint8_t domainName [256]`

### 8.1060.1 Detailed Description

This structure contains the DomainName Information

#### Parameters

<i>domainLen</i>	<ul style="list-style-type: none"> <li>• length of the recieved <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>

## 8.1060.2 Field Documentation

8.1060.2.1 `uint16_t wds_Domain::domainLen`

8.1060.2.2 `uint8_t wds_Domain::domainName[256]`

## 8.1061 wds\_DomainNameList Struct Reference

### Data Fields

- `uint8_t numInstances`
- struct `wds_Domain domain` [10]

### 8.1061.1 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>

## 8.1061.2 Field Documentation

8.1061.2.1 `struct wds_Domain wds_DomainNameList::domain[10]`

8.1061.2.2 `uint8_t wds_DomainNameList::numInstances`

## 8.1062 wds\_GPRSQoS Struct Reference

### Data Fields

- `uint32_t precedenceClass`
- `uint32_t delayClass`
- `uint32_t reliabilityClass`
- `uint32_t peakThroughputClass`
- `uint32_t meanThroughputClass`

### 8.1062.1 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>

## 8.1062.2 Field Documentation

8.1062.2.1 `uint32_t wds_GPRSQoS::delayClass`

8.1062.2.2 `uint32_t wds_GPRSQoS::meanThroughputClass`

8.1062.2.3 `uint32_t wds_GPRSQoS::peakThroughputClass`

8.1062.2.4 `uint32_t wds_GPRSQoS::precedenceClass`

8.1062.2.5 `uint32_t wds_GPRSQoS::reliabilityClass`

## 8.1063 wds\_IPV6AddressInfo Struct Reference

### Data Fields

- `uint8_t` [IPV6PrefixLen](#)
- `uint16_t` [IPAddressV6](#) [8]

### 8.1063.1 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>

## 8.1063.2 Field Documentation

8.1063.2.1 `uint16_t wds_IPV6AddressInfo::IPAddressV6[8]`

8.1063.2.2 `uint8_t wds_IPV6AddressInfo::IPV6PrefixLen`

## 8.1064 wds\_IPV6GWAddressInfo Struct Reference

## Data Fields

- uint8\_t [gwV6PrefixLen](#)
- uint16\_t [gwAddressV6](#) [8]

### 8.1064.1 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>

### 8.1064.2 Field Documentation

8.1064.2.1 uint16\_t wds\_IPV6GWAddressInfo::gwAddressV6[8]

8.1064.2.2 uint8\_t wds\_IPV6GWAddressInfo::gwV6PrefixLen

## 8.1065 wds\_PCSCFFQDNAddress Struct Reference

## Data Fields

- uint16\_t [fqdnLen](#)
- uint8\_t [fqdnAddr](#) [256]

### 8.1065.1 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>

### 8.1065.2 Field Documentation

8.1065.2.1 uint8\_t wds\_PCSCFFQDNAddress::fqdnAddr[256]

8.1065.2.2 uint16\_t wds\_PCSCFFQDNAddress::fqdnLen

## 8.1066 wds\_PCSCFFQDNAddressList Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- struct [wds\\_PCSCFFQDNAddress](#) [pcsffQDNAddress](#) [10]

### 8.1066.1 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>

### 8.1066.2 Field Documentation

8.1066.2.1 uint8\_t wds\_PCSCFFQDNAddressList::numInstances

8.1066.2.2 struct wds\_PCSCFFQDNAddress wds\_PCSCFFQDNAddressList::pcsffQDNAddress[10]

## 8.1067 wds\_PCSCFIPv4ServerAddressList Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- uint32\_t [pccsfIPv4Addr](#) [64]

### 8.1067.1 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>pccsfIPv4Addr</i>	<ul style="list-style-type: none"> <li>• P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)</li> </ul>

### 8.1067.2 Field Documentation

8.1067.2.1 uint8\_t wds\_PCSCFIPv4ServerAddressList::numInstances

8.1067.2.2 uint32\_t wds\_PCSCFIPv4ServerAddressList::pccsfIPv4Addr[64]

## 8.1068 wds\_ProfileIdentifier Struct Reference

## Data Fields

- uint8\_t [profileType](#)
- uint8\_t [profileIndex](#)

### 8.1068.1 Detailed Description

This structure contains the Profile Identifier Information

#### 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>

### 8.1068.2 Field Documentation

8.1068.2.1 uint8\_t wds\_ProfileIdentifier::profileIndex

8.1068.2.2 uint8\_t wds\_ProfileIdentifier::profileType

## 8.1069 wds\_profileInfo Union Reference

## Data Fields

- [LibPackprofile\\_3GPP](#) SlqsProfile3GPP
- [LibPackprofile\\_3GPP2](#) SlqsProfile3GPP2

### 8.1069.1 Detailed Description

This union consist of profile\_3GPP and profile\_3GPP2 out of which one will be used to create profile.

### 8.1069.2 Field Documentation

8.1069.2.1 LibPackprofile\_3GPP wds\_profileInfo::SlqsProfile3GPP

8.1069.2.2 LibPackprofile\_3GPP2 wds\_profileInfo::SlqsProfile3GPP2

## 8.1070 wds\_UMTSMinQoS Struct Reference

## Data Fields

- uint8\_t [trafficClass](#)
- uint32\_t [maxUplinkBitrate](#)
- uint32\_t [maxDownlinkBitrate](#)
- uint32\_t [grntUplinkBitrate](#)
- uint32\_t [grntDownlinkBitrate](#)

- uint8\_t [qosDeliveryOrder](#)
- uint32\_t [maxSDUSize](#)
- uint8\_t [sduErrorRatio](#)
- uint8\_t [resBerRatio](#)
- uint8\_t [deliveryErrSDU](#)
- uint32\_t [transferDelay](#)
- uint32\_t [trafficPriority](#)

### 8.1070.1 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 \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 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>	<ul style="list-style-type: none"> <li>- delivery of erroneous SDUs</li> <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>	<ul style="list-style-type: none"> <li>- Transfer delay (ms)</li> <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>	<ul style="list-style-type: none"> <li>- Transfer handling priority</li> <li>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

## 8.1070.2 Field Documentation

8.1070.2.1 `uint8_t wds_UMTSMinQoS::deliveryErrSDU`

8.1070.2.2 `uint32_t wds_UMTSMinQoS::grntDownlinkBitrate`

8.1070.2.3 `uint32_t wds_UMTSMinQoS::grntUplinkBitrate`

8.1070.2.4 `uint32_t wds_UMTSMinQoS::maxDownlinkBitrate`

8.1070.2.5 `uint32_t wds_UMTSMinQoS::maxSDUSize`

8.1070.2.6 `uint32_t wds_UMTSMinQoS::maxUplinkBitrate`

8.1070.2.7 `uint8_t wds_UMTSMinQoS::qosDeliveryOrder`

8.1070.2.8 `uint8_t wds_UMTSMInQoS::resBerRatio`

8.1070.2.9 `uint8_t wds_UMTSMInQoS::sduErrorRatio`

8.1070.2.10 `uint8_t wds_UMTSMInQoS::trafficClass`

8.1070.2.11 `uint32_t wds_UMTSMInQoS::trafficPriority`

8.1070.2.12 `uint32_t wds_UMTSMInQoS::transferDelay`

## 8.1071 WdsByteTotals Struct Reference

### Data Fields

- [ULONG](#) \* `pV4sessionId`
- [ULONG](#) \* `pV6sessionId`
- struct [WdsByteTotalsElmnts](#) `ByteTotalsElmntsV4`
- struct [WdsByteTotalsElmnts](#) `ByteTotalsElmntsV6`

### 8.1071.1 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.

### 8.1071.2 Field Documentation

8.1071.2.1 `struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV4`

8.1071.2.2 `struct WdsByteTotalsElmnts WdsByteTotals::ByteTotalsElmntsV6`

8.1071.2.3 `ULONG*` `WdsByteTotals::pV4sessionId`

8.1071.2.4 `ULONG*` `WdsByteTotals::pV6sessionId`

## 8.1072 WdsByteTotalsElmnts Struct Reference

### Data Fields

- [ULONGLONG](#) \* [pTXTotalBytes](#)
- [ULONGLONG](#) \* [pRXTotalBytes](#)

### 8.1072.1 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>

### 8.1072.2 Field Documentation

8.1072.2.1 [ULONGLONG](#)\* WdsByteTotalsElmnts::pRXTotalBytes

8.1072.2.2 [ULONGLONG](#)\* WdsByteTotalsElmnts::pTXTotalBytes

## 8.1073 WdsClientLeaseChange Struct Reference

### Data Fields

- [BYTE](#) \* [pEnableNotification](#)

### 8.1073.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

#### Parameters

<i>pEnable-Notification</i>	[IN] <ul style="list-style-type: none"> <li>• Enable Notification or not</li> </ul>
-----------------------------	--

### 8.1073.2 Field Documentation

8.1073.2.1 [BYTE](#)\* WdsClientLeaseChange::pEnableNotification

## 8.1074 WdsConnectionRate Struct Reference

### Data Fields

- [ULONG](#) \* [pV4sessionId](#)
- [ULONG](#) \* [pV6sessionId](#)

- struct [WdsConnectionRateElmnts ConnRateElmntsV4](#)
- struct [WdsConnectionRateElmnts ConnRateElmntsV6](#)

### 8.1074.1 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.

### 8.1074.2 Field Documentation

8.1074.2.1 struct `WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV4`

8.1074.2.2 struct `WdsConnectionRateElmnts WdsConnectionRate::ConnRateElmntsV6`

8.1074.2.3 `ULONG*` `WdsConnectionRate::pV4sessionId`

8.1074.2.4 `ULONG*` `WdsConnectionRate::pV6sessionId`

## 8.1075 WdsConnectionRateElmnts Struct Reference

### Data Fields

- `ULONG *` `pCurrentChannelTXRate`
- `ULONG *` `pCurrentChannelRXRate`
- `ULONG *` `pMaxChannelTXRate`
- `ULONG *` `pMaxChannelRXRate`

### 8.1075.1 Detailed Description

WDS Connection rates request data structure for individual session

## Parameters

<i>pCurrent-ChannelTX-Rate[OUT]</i>	<ul style="list-style-type: none"> <li>Instantaneous channel Tx rate in bits per second.</li> </ul>
<i>pCurrent-ChannelRX-Rate[OUT]</i>	<ul style="list-style-type: none"> <li>Instantaneous channel Rx rate in bits per second.</li> </ul>
<i>pMaxChannelTXRate[OUT]</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[OUT]</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>

## 8.1075.2 Field Documentation

8.1075.2.1 **ULONG\*** WdsConnectionRateElmnts::pCurrentChannelRXRate8.1075.2.2 **ULONG\*** WdsConnectionRateElmnts::pCurrentChannelTXRate8.1075.2.3 **ULONG\*** WdsConnectionRateElmnts::pMaxChannelRXRate8.1075.2.4 **ULONG\*** WdsConnectionRateElmnts::pMaxChannelTXRate

## 8.1076 WdsDHCPv4ClientLeaseInd Struct Reference

## Data Fields

- [WdsDHCPv4ProfileId](#) \* [pProfileId](#)
- BYTE** \* [pLeaseState](#)
- ULONG** \* [pIPv4Addr](#)
- [DHCPOptionList](#) \* [pOptList](#)

## 8.1076.1 Detailed Description

This structure contains DHCPv4 client lease status

## Parameters

<i>pProfileId</i>	<ul style="list-style-type: none"> <li>• Profile Type and Id</li> </ul>
<i>pLeaseState</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– 0 - active, newly acquired</li> <li>– 1 - active, renewed</li> <li>– 2 - active, renewing</li> <li>– 3 - active, rebinding</li> <li>– 4 - inactive, expired</li> <li>– 5 - inactive, renew refused</li> <li>– 6 - inactive, rebind refused</li> <li>– 7 - inactive, other</li> </ul> </li> </ul>
<i>pIPv4Addr</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– IPv4 Address</li> </ul> </li> </ul>
<i>pOptList</i>	<ul style="list-style-type: none"> <li>• Option list</li> </ul>

## 8.1076.2 Field Documentation

8.1076.2.1 **ULONG\*** WdsDHCPv4ClientLeaseInd::pIPv4Addr8.1076.2.2 **BYTE\*** WdsDHCPv4ClientLeaseInd::pLeaseState8.1076.2.3 **DHCPOptionList\*** WdsDHCPv4ClientLeaseInd::pOptList8.1076.2.4 **WdsDHCPv4ProfileId\*** WdsDHCPv4ClientLeaseInd::pProfileId

## 8.1077 WdsDHCPv4Config Struct Reference

## Data Fields

- [WdsDHCPv4ProfileId](#) \* [pProfileId](#)
- [WdsDHCPv4HWConfig](#) \* [pHWConfig](#)
- [WdsDHCPv4OptionList](#) \* [pRequestOptionList](#)

## 8.1077.1 Detailed Description

WDS SWI DHCPv4 Config Structure

## Parameters

<i>pProfileId</i>	<b>[IN]</b> <ul style="list-style-type: none"> <li>• pointer to Profile Id structure</li> </ul>
<i>pHWConfig</i>	<b>[IN/OUT]</b> <ul style="list-style-type: none"> <li>• pointer to HW Config structure</li> </ul>

<i>pRequestOption-List</i>	[IN/OUT] <ul style="list-style-type: none"> <li>• pointer to Option List structure to be sent in DHCP request</li> </ul>
----------------------------	--

### 8.1077.2 Field Documentation

8.1077.2.1 WdsDHCPv4HWConfig\* WdsDHCPv4Config::pHwConfig

8.1077.2.2 WdsDHCPv4ProfileId\* WdsDHCPv4Config::pProfileId

8.1077.2.3 WdsDHCPv4OptionList\* WdsDHCPv4Config::pRequestOptionList

## 8.1078 wdsDhcpv4HwConfig Struct Reference

### Data Fields

- uint8\_t [hwType](#)
- uint8\_t [chaddrLen](#)
- uint8\_t [chaddr](#) [16]

### 8.1078.1 Detailed Description

#### Parameters

<i>hwType</i>	DHCP HW Type, examples: <ul style="list-style-type: none"> <li>• 0 - Ethernet</li> <li>• 20 - Serial</li> </ul>
<i>chaddrLen</i>	Length of chaddr field, examples: <ul style="list-style-type: none"> <li>• 6 for Ethernet MAC address</li> </ul>
<i>chaddr</i>	Client hardware address

### 8.1078.2 Field Documentation

8.1078.2.1 uint8\_t wdsDhcpv4HwConfig::chaddr[16]

8.1078.2.2 uint8\_t wdsDhcpv4HwConfig::chaddrLen

8.1078.2.3 uint8\_t wdsDhcpv4HwConfig::hwType

## 8.1079 WdsDHCPv4HWConfig Struct Reference

### Data Fields

- BYTE [hwType](#)
- BYTE [chaddrLen](#)
- BYTE [chaddr](#) [16]

### 8.1079.1 Detailed Description

WDS SWI DHCPv4 HW Config Structure.

#### Parameters

<i>hwType</i>	<ul style="list-style-type: none"> <li>• HW Type 1 - Ethernet 20 - Serial</li> </ul>
<i>chaddrlen</i>	<ul style="list-style-type: none"> <li>• chaddrlen</li> </ul>
<i>chaddr</i>	<ul style="list-style-type: none"> <li>• chaddr. Max size 16 bytes</li> </ul>

### 8.1079.2 Field Documentation

8.1079.2.1 **BYTE** WdsDHCPv4HWConfig::chaddr[16]

8.1079.2.2 **BYTE** WdsDHCPv4HWConfig::chaddrLen

8.1079.2.3 **BYTE** WdsDHCPv4HWConfig::hwType

## 8.1080 WdsDHCPv4Option Struct Reference

#### Data Fields

- [BYTE optCode](#)
- [BYTE optValLen](#)
- [BYTE optVal](#) [255]

### 8.1080.1 Detailed Description

WDS SWI DHCPv4 Option Structure

#### Parameters

<i>optCode</i>	<ul style="list-style-type: none"> <li>• Option code <ul style="list-style-type: none"> <li>– 0 - 255</li> </ul> </li> </ul>
<i>optValLen</i>	<ul style="list-style-type: none"> <li>• Option value length <ul style="list-style-type: none"> <li>– 0 - 255</li> </ul> </li> </ul>
<i>optVal</i>	<ul style="list-style-type: none"> <li>• Option value</li> </ul>

### 8.1080.2 Field Documentation

8.1080.2.1 **BYTE** WdsDHCPv4Option::optCode

8.1080.2.2 `BYTE WdsDhcpv4Option::optVal[255]`

8.1080.2.3 `BYTE WdsDhcpv4Option::optValLen`

## 8.1081 wdsDhcpv4Option Struct Reference

### Data Fields

- `uint8_t optCode`
- `uint8_t optValLen`
- `uint8_t optVal [255]`

### 8.1081.1 Detailed Description

#### Parameters

<i>optCode</i>	Option code <ul style="list-style-type: none"><li>• 0 - 255</li></ul>
<i>optValLen</i>	Option value length <ul style="list-style-type: none"><li>• 0 - 255</li></ul>
<i>optVal</i>	Option Value

### 8.1081.2 Field Documentation

8.1081.2.1 `uint8_t wdsDhcpv4Option::optCode`

8.1081.2.2 `uint8_t wdsDhcpv4Option::optVal[255]`

8.1081.2.3 `uint8_t wdsDhcpv4Option::optValLen`

## 8.1082 wdsDhcpv4OptionList Struct Reference

### Data Fields

- `uint8_t numOpt`
- `wdsDhcpv4Option * pOptList`

### 8.1082.1 Detailed Description

#### Parameters

<i>numOpt</i>	number of options <ul style="list-style-type: none"><li>• 0 - 255</li></ul>
<i>pOptList</i>	pointer to list of DHCP Options

### 8.1082.2 Field Documentation

8.1082.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`

8.1082.2.2 wdsDhcpv4Option\* wdsDhcpv4OptionList::pOptList

## 8.1083 WdsDHCPv4OptionList Struct Reference

### Data Fields

- [BYTE numOpt](#)
- [WdsDHCPv4Option \\* pOptList](#)

### 8.1083.1 Detailed Description

WDS SWI DHCPv4 Option List Structure

#### Parameters

<i>numOpt</i>	<ul style="list-style-type: none"> <li>• number of options <ul style="list-style-type: none"> <li>– 0 - 255</li> </ul> </li> </ul>
<i>pOptList</i>	<ul style="list-style-type: none"> <li>• pointer to list of DHCP Options</li> </ul>

### 8.1083.2 Field Documentation

8.1083.2.1 [BYTE WdsDHCPv4OptionList::numOpt](#)

8.1083.2.2 [WdsDHCPv4Option\\* WdsDHCPv4OptionList::pOptList](#)

## 8.1084 WdsDHCPv4ProfileId Struct Reference

### Data Fields

- [BYTE profileType](#)
- [BYTE profileId](#)

### 8.1084.1 Detailed Description

WDS SWI DHCPv4 Profile Identifier Structure

#### Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>• 0 for 3GPP</li> </ul>
<i>profileId</i>	<ul style="list-style-type: none"> <li>• 1 to 24 for 3GPP profile</li> </ul>

### 8.1084.2 Field Documentation

8.1084.2.1 **BYTE** WdsDHCPv4ProfileId::profileId

8.1084.2.2 **BYTE** WdsDHCPv4ProfileId::profileType

## 8.1085 wdsDhcpv4ProfileId Struct Reference

### Data Fields

- `uint8_t` [profileType](#)
- `uint8_t` [profileId](#)

### 8.1085.1 Detailed Description

#### Parameters

<i>profileType</i>	profile type <ul style="list-style-type: none"><li>• 0 - 3GPP</li></ul>
<i>profileId</i>	profile index <ul style="list-style-type: none"><li>• index identifying the profile 1-24 valid for 3GPP profile type (EM74xx and onwards)</li></ul>

### 8.1085.2 Field Documentation

8.1085.2.1 `uint8_t` wdsDhcpv4ProfileId::profileId

8.1085.2.2 `uint8_t` wdsDhcpv4ProfileId::profileType

## 8.1086 WDSGetLoopbackData Struct Reference

### Data Fields

- **BYTE** [ByteLoopbackMode](#)
- **BYTE** [ByteLoopbackMultiplier](#)

### 8.1086.1 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>

### 8.1086.2 Field Documentation

8.1086.2.1 **BYTE** WDSGetLoopbackData::ByteLoopbackMode

8.1086.2.2 **BYTE** WDSGetLoopbackData::ByteLoopbackMultiplier

## 8.1087 WdslpAddressInfoReq Struct Reference

#### Data Fields

- [ULONG](#) \* [pv4sessionId](#)
- [ULONG](#) \* [pv6sessionId](#)
- [QmiWdslpAddressInfo](#) [ip](#)

### 8.1087.1 Field Documentation

8.1087.1.1 **QmiWdslpAddressInfo** WdslpAddressInfoReq::ip

8.1087.1.2 **ULONG\*** WdslpAddressInfoReq::pv4sessionId

8.1087.1.3 **ULONG\*** WdslpAddressInfoReq::pv6sessionId

## 8.1088 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](#)

### 8.1088.1 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>

### 8.1088.2 Field Documentation

8.1088.2.1 **ULONG\*** WdsPktStatisticsElmnts::pRXDroppedCount

8.1088.2.2 **ULONGLONG\*** WdsPktStatisticsElmnts::pRXOkBytesCount

8.1088.2.3 **ULONGLONG\*** WdsPktStatisticsElmnts::pRXOKBytesLastCall

- 8.1088.2.4 **ULONG\*** WdsPktStatisticsElmnts::pRXPacketErrors
- 8.1088.2.5 **ULONG\*** WdsPktStatisticsElmnts::pRXPacketOverflows
- 8.1088.2.6 **ULONG\*** WdsPktStatisticsElmnts::pRXPacketSuccesses
- 8.1088.2.7 **ULONG\*** WdsPktStatisticsElmnts::pTXDroppedCount
- 8.1088.2.8 **ULONGLONG\*** WdsPktStatisticsElmnts::pTXOkBytesCount
- 8.1088.2.9 **ULONGLONG\*** WdsPktStatisticsElmnts::pTXOkBytesLastCall
- 8.1088.2.10 **ULONG\*** WdsPktStatisticsElmnts::pTXPacketErrors
- 8.1088.2.11 **ULONG\*** WdsPktStatisticsElmnts::pTXPacketOverflows
- 8.1088.2.12 **ULONG\*** WdsPktStatisticsElmnts::pTXPacketSuccesses

## 8.1089 WdsPktStatisticsReq Struct Reference

### Data Fields

- **ULONG \*** [pStatMask](#)

### 8.1089.1 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>
------------------	--

### 8.1089.2 Field Documentation

- 8.1089.2.1 **ULONG\*** WdsPktStatisticsReq::pStatMask

## 8.1090 WdsPktStatisticsResp Struct Reference

### Data Fields

- **ULONG \*** [pV4sessionId](#)
- **ULONG \*** [pV6sessionId](#)
- struct [WdsPktStatisticsElmnts PktStatElmntsV4](#)
- struct [WdsPktStatisticsElmnts PktStatElmntsV6](#)

### 8.1090.1 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.

## 8.1090.2 Field Documentation

8.1090.2.1 struct `WdsPktStatisticsElmnts` `WdsPktStatisticsResp::PktStatElmntsV4`

8.1090.2.2 struct `WdsPktStatisticsElmnts` `WdsPktStatisticsResp::PktStatElmntsV6`

8.1090.2.3 `ULONG*` `WdsPktStatisticsResp::pV4sessionId`

8.1090.2.4 `ULONG*` `WdsPktStatisticsResp::pV6sessionId`

## 8.1091 WdsProfileParam Union Reference

## Data Fields

- struct [Profile3GPP](#) `SlqsProfile3GPP`
- struct [Profile3GPP2](#) `SlqsProfile3GPP2`

## 8.1091.1 Detailed Description

This union [WdsProfileParam](#) consist of [Profile3GPP](#) and [Profile3GPP2](#) out of which one will be used to create profile.

## 8.1091.2 Field Documentation

8.1091.2.1 struct `Profile3GPP` `WdsProfileParam::SlqsProfile3GPP`

8.1091.2.2 struct `Profile3GPP2` `WdsProfileParam::SlqsProfile3GPP2`

## 8.1092 WdsRunTimeSettings Struct Reference

## Data Fields

- `ULONG` \* `v4sessionId`

- [ULONG](#) \* [v6sessionId](#)
- struct [qmiWdsRunTimeSettings](#) [rts](#)

### 8.1092.1 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.

### 8.1092.2 Field Documentation

8.1092.2.1 struct [qmiWdsRunTimeSettings](#) [WdsRunTimeSettings::rts](#)

8.1092.2.2 [ULONG](#)\* [WdsRunTimeSettings::v4sessionId](#)

8.1092.2.3 [ULONG](#)\* [WdsRunTimeSettings::v6sessionId](#)

## 8.1093 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](#)

### 8.1093.1 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.

### 8.1093.2 Field Documentation

- 8.1093.2.1 **BYTE\*** wdsSetEventReportReq::pCurrChannelRateInd
- 8.1093.2.2 **BYTE\*** wdsSetEventReportReq::pCurrDataBearerTechInd
- 8.1093.2.3 **BYTE\*** wdsSetEventReportReq::pCurrPrefDataSysInd
- 8.1093.2.4 **BYTE\*** wdsSetEventReportReq::pDataBearerTechInd
- 8.1093.2.5 **BYTE\*** wdsSetEventReportReq::pDataCallStatusChangeInd
- 8.1093.2.6 **BYTE\*** wdsSetEventReportReq::pDataSystemStatusChangeInd
- 8.1093.2.7 **BYTE\*** wdsSetEventReportReq::pDormancyStatusInd
- 8.1093.2.8 **BYTE\*** wdsSetEventReportReq::pEVDOPageMonPerChangeInd
- 8.1093.2.9 **BYTE\*** wdsSetEventReportReq::pMIPStatusInd
- 8.1093.2.10 **TrStatInd\*** wdsSetEventReportReq::pTransferStatInd

## 8.1094 WDSSetLoopbackData Struct Reference

### Data Fields

- **BYTE\*** pLoopbackMode
- **BYTE\*** pLoopbackMultiplier

### 8.1094.1 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>

### 8.1094.2 Field Documentation

- 8.1094.2.1 **BYTE\*** WDSSetLoopbackData::pLoopbackMode
- 8.1094.2.2 **BYTE\*** WDSSetLoopbackData::pLoopbackMultiplier

## 8.1095 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](#)

### 8.1095.1 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>

### 8.1095.2 Field Documentation

8.1095.2.1 unsigned long WDSSWICurrentChannelRates::current\_channel\_rx\_rate

8.1095.2.2 unsigned long WDSSWICurrentChannelRates::current\_channel\_tx\_rate

8.1095.2.3 unsigned long WDSSWICurrentChannelRates::max\_channel\_rx\_rate

8.1095.2.4 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 common.h File Reference

#### Data Structures

- struct [pack\\_qmi\\_t](#)
- struct [unpack\\_qmi\\_t](#)

#### Macros

- #define [SDU\\_HDR\\_LEN](#) (3)
- #define [MINREQBKLEN](#) (2048)
- #define [MSGID\\_AND\\_LEN](#) (4)
- #define [MSGID\\_DONT\\_CARE](#) (0xffff)
- #define [UNUSEDPARAM](#)(x) (void)x
- #define [DEAULT\\_LOC\\_TIMEOUT\\_IN\\_SEC](#) 2
- #define [SDK\\_VALIDATE\\_INPUT\\_PACK\\_PARAM](#)(pCtx, pBuf, pLen)

#### Typedefs

- typedef void(\* [logger](#) )(uint8\_t lvl, const char \*buff)

## Enumerations

- enum [eLOG\\_LEVEL](#) {  
[eLOG\\_INFO](#),  
[eLOG\\_DEBUG](#),  
[eLOG\\_WARN](#),  
[eLOG\\_FATAL](#) }
- enum [eTimeout](#) {  
[eTIMEOUT\\_2\\_S](#) = 2000,  
[eTIMEOUT\\_5\\_S](#) = 5000,  
[eTIMEOUT\\_8\\_S](#) = 8000,  
[eTIMEOUT\\_10\\_S](#) = 10000,  
[eTIMEOUT\\_20\\_S](#) = 20000,  
[eTIMEOUT\\_30\\_S](#) = 30000,  
[eTIMEOUT\\_60\\_S](#) = 60000,  
[eTIMEOUT\\_300\\_S](#) = 300000,  
[eTIMEOUT\\_DEFAULT](#) = [eTIMEOUT\\_8\\_S](#) }
- enum [eQMI\\_SVC](#) {  
[eCTL](#),  
[eWDS](#),  
[eDMS](#),  
[eNAS](#) =3,  
[eQOS](#),  
[eSMS](#) =5,  
[eUIM](#) =0x0B,  
[eLOC](#) =0x10,  
[eTMD](#) =0x18,  
[eSWIOMA](#) =240,  
[eSWILOC](#) =246 }
- enum [msgtype](#) {  
[eREQ](#) =0,  
[eRSP](#) =2,  
[eIND](#) =4 }

## Functions

- [uint16\\_t helper\\_get\\_xid](#) ([uint8\\_t](#) \*qmi\_resp)
- [const char \\*](#) [helper\\_get\\_resp\\_ctx](#) ([uint8\\_t](#) svc, [uint8\\_t](#) \*pbuf, [uint32\\_t](#) len, [unpack\\_qmi\\_t](#) \*pCtx)
- [unsigned](#) [unpack\\_result\\_code\\_only](#) ([uint8\\_t](#) \*pMdmResp)
- [int](#) [helper\\_set\\_log\\_func](#) ([logger](#) func)
- [void](#) [libpack\\_log](#) ([uint8\\_t](#) lvl, [const char](#) \*fmt,...)
- [int](#) [helper\\_set\\_log\\_lvl](#) ([uint8\\_t](#) lvl)
- [void](#) [fill\\_sdu\\_hdr](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf)
- [void](#) [fill\\_pack\\_ctx](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [uint8\\_t](#) svc, [int](#) timeout)
- [char \\*](#) [get\\_version](#) ()
- [char \\*](#) [libpack\\_GetVersion](#) ()

## Variables

- [logger](#) [glog](#)
- [uint8\\_t](#) [gloglvl](#)

## 9.2.1 Macro Definition Documentation

9.2.1.1 `#define DEAUULT_LOC_TIMEOUT_IN_SEC 2`

9.2.1.2 `#define MINREQBKLEN (2048)`

9.2.1.3 `#define MSGID_AND_LEN (4)`

9.2.1.4 `#define MSGID_DONT_CARE (0xffff)`

9.2.1.5 `#define SDK_VALIDATE_INPUT_PACK_PARAM( pCtx, pBuf, pLen )`

**Value:**

```
if ( (pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
{ \
    libpack_log(eLOG_DEBUG, "[ pack] %s parameter NULL\n", __func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
}
```

9.2.1.6 `#define SDU_HDR_LEN (3)`

9.2.1.7 `#define UNUSEDPARAM( x ) (void)x`

## 9.2.2 Typedef Documentation

9.2.2.1 `typedef void(* logger)(uint8_t lvl, const char *buff)`

## 9.2.3 Enumeration Type Documentation

9.2.3.1 `enum eLOG_LEVEL`

log levels

Enumerator

***eLOG\_INFO***  
***eLOG\_DEBUG***  
***eLOG\_WARN***  
***eLOG\_FATAL***

9.2.3.2 `enum eQMI_SVC`

qmi service

Enumerator

***eCTL***  
***eWDS***  
***eDMS***  
***eNAS***  
***eQOS***  
***eSMS***  
***eUIM***

***eLOC***  
***eTMD***  
***eSWIOMA***  
***eSWILOC***

### 9.2.3.3 enum eTimeout

eTimeout

Enumerator

***eTIMEOUT\_2\_S***  
***eTIMEOUT\_5\_S***  
***eTIMEOUT\_8\_S***  
***eTIMEOUT\_10\_S***  
***eTIMEOUT\_20\_S***  
***eTIMEOUT\_30\_S***  
***eTIMEOUT\_60\_S***  
***eTIMEOUT\_300\_S***  
***eTIMEOUT\_DEFAULT***

### 9.2.3.4 enum msgtype

qmi message type

Enumerator

***eREQ***  
***eRSP***  
***eIND***

## 9.2.4 Function Documentation

9.2.4.1 void fill\_pack\_ctx ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, uint8\_t svc, int timeout )

9.2.4.2 void fill\_sdu\_hdr ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf )

9.2.4.3 char\* get\_version ( )

Returns

version string

9.2.4.4 const char\* helper\_get\_resp\_ctx ( uint8\_t svc, uint8\_t \* pbuf, uint32\_t len, unpack\_qmi\_t \* pCtx )

extract msgid/xid/type from modem reply

Parameters

in	svc	qmi service
in	pbuf	qmi response/indication
in	len	response/indication length
out	pCtx	unpacked context

## Returns

qmi message string

9.2.4.5 `uint16_t helper_get_xid ( uint8_t * qmi_resp )`

9.2.4.6 `int helper_set_log_func ( logger func )`

set log function

9.2.4.7 `int helper_set_log_lvl ( uint8_t lvl )`

set log level

9.2.4.8 `char* libpack_GetVersion ( )`

## Returns

version string

9.2.4.9 `void libpack_log ( uint8_t lvl, const char * fmt, ... )`

9.2.4.10 `unsigned unpack_result_code_only ( uint8_t * pMdmResp )`

common handler for unpacking response with TLV type 0x02 only

## 9.2.5 Variable Documentation

9.2.5.1 `logger glog`

9.2.5.2 `uint8_t gloglvl`

## 9.3 dms.h File Reference

### Data Structures

- struct [unpack\\_dms\\_GetModelID\\_t](#)
- struct [unpack\\_dms\\_GetIMSI\\_t](#)
- struct [unpack\\_dms\\_GetFirmwareInfo\\_t](#)
- struct [unpack\\_dms\\_GetPower\\_t](#)
- struct [unpack\\_dms\\_GetSerialNumbers\\_t](#)
- struct [unpack\\_dms\\_GetHardwareRevision\\_t](#)
- struct [unpack\\_dms\\_SLQSGetBandCapability\\_t](#)
- struct [unpack\\_dms\\_GetDeviceCapabilities\\_t](#)
- struct [unpack\\_dms\\_GetFirmwareRevisions\\_t](#)
- struct [unpack\\_dms\\_GetFirmwareRevision\\_t](#)
- struct [unpack\\_dms\\_GetDeviceSerialNumbers\\_t](#)
- struct [unpack\\_dms\\_GetPRLVersion\\_t](#)
- struct [unpack\\_dms\\_GetNetworkTime\\_t](#)
- struct [unpack\\_dms\\_GetVoiceNumber\\_t](#)
- struct [unpack\\_dms\\_GetDeviceHardwareRev\\_t](#)
- struct [unpack\\_dms\\_GetFSN\\_t](#)

- struct [unpack\\_dms\\_GetDeviceCap\\_t](#)
- struct [pack\\_dms\\_SetPower\\_t](#)
- struct [unpack\\_dms\\_SetPower\\_t](#)
- struct [unpack\\_dms\\_GetBandCapability\\_t](#)
- struct [unpack\\_dms\\_GetUSBComp\\_t](#)
- struct [pack\\_dms\\_SetUSBComp\\_t](#)
- struct [unpack\\_dms\\_SetUSBComp\\_t](#)
- struct [pack\\_dms\\_SetCustFeature\\_t](#)
- struct [unpack\\_dms\\_SetCustFeature\\_t](#)
- struct [unpack\\_dms\\_GetCustFeature\\_t](#)
- struct [unpack\\_dms\\_SetFirmwarePreference\\_t](#)
- struct [unpack\\_dms\\_GetCrashAction\\_t](#)
- struct [pack\\_dms\\_SetCrashAction\\_t](#)
- struct [unpack\\_dms\\_SetCrashAction\\_t](#)
- struct [unpack\\_dms\\_GetDeviceMfr\\_t](#)
- struct [pack\\_dms\\_SetEventReport\\_t](#)
- struct [unpack\\_dms\\_SetEventReport\\_t](#)
- struct [dms\\_OperatingModeTlv](#)
- struct [dms\\_ActivationStatusTlv](#)
- struct [unpack\\_dms\\_SetEventReport\\_ind\\_t](#)
- struct [pack\\_dms\\_UIMGetICCID\\_t](#)
- struct [unpack\\_dms\\_UIMGetICCID\\_t](#)
- struct [pack\\_dms\\_SetCustFeaturesV2\\_t](#)
- struct [unpack\\_dms\\_SetCustFeaturesV2\\_t](#)
- struct [pack\\_dms\\_GetCustFeaturesV2\\_t](#)
- struct [DMSgetCustomInput](#)
- struct [DMScustSettingInfo](#)
- struct [DMScustSettingList](#)
- struct [DMSgetCustomFeatureV2](#)
- struct [unpack\\_dms\\_GetCustFeaturesV2\\_t](#)
- struct [unpack\\_dms\\_GetActivationState\\_t](#)
- struct [image\\_info\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetFirmwareCurr\\_t](#)
- struct [pack\\_dms\\_SLQSSwiSetDyingGaspCfg\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiSetDyingGaspCfg\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiClearDyingGaspStatistics\\_t](#)
- struct [packgetDyingGaspStatistics](#)
- struct [unpack\\_dms\\_SLQSSwiGetDyingGaspStatistics\\_t](#)
- struct [packgetDyingGaspCfg](#)
- struct [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg\\_t](#)
- struct [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_Ind\\_t](#)
- struct [pack\\_dms\\_SLQSDmsSwiIndicationRegister\\_t](#)
- struct [unpack\\_dms\\_SLQSDmsSwiIndicationRegister\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetFwUpdateStatus\\_t](#)

## Macros

- #define [DMS\\_UINT8\\_MAX\\_STRING\\_SZ](#) 255
- #define [DMS\\_MAX\\_CUST\\_ID\\_LEN](#) 64
- #define [DMS\\_MAX\\_CUST\\_VALUE\\_LEN](#) 8
- #define [DMS\\_IMGDETAILS\\_LEN](#) 16
- #define [DMS\\_MAX\\_FWUPDATE\\_LOG\\_STR\\_SZ](#) 255
- #define [DMS\\_MAX\\_FWUPDATE\\_REF\\_STR\\_SZ](#) 15
- #define [DMS\\_SLQSFWINFO\\_MODELID\\_SZ](#) 20
- #define [DMS\\_SLQSFWINFO\\_BOOTVERSION\\_SZ](#) 85
- #define [DMS\\_SLQSFWINFO\\_APPVERSION\\_SZ](#) 85
- #define [DMS\\_SLQSFWINFO\\_SKU\\_SZ](#) 15
- #define [DMS\\_SLQSFWINFO\\_PACKAGEID\\_SZ](#) 85
- #define [DMS\\_SLQSFWINFO\\_CARRIER\\_SZ](#) 20
- #define [DMS\\_SLQSFWINFO\\_PRIVERSION\\_SZ](#) 16
- #define [DMS\\_SLQSFWINFO\\_CUR\\_CARR\\_NAME](#) 17
- #define [DMS\\_SLQSFWINFO\\_CUR\\_CARR\\_REV](#) 13
- #define [MAX\\_BUILD\\_ID\\_LEN](#) 255
- #define [UNIQUE\\_ID\\_LEN](#) 16
- #define [SLQS\\_MAX\\_DYING\\_GASP\\_CFG\\_SMS\\_CONTENT\\_LENGTH](#) 160
- #define [SLQS\\_MAX\\_DYING\\_GASP\\_CFG\\_SMS\\_NUMBER\\_LENGTH](#) 20
- #define [DMS\\_PM\\_ONLINE](#) 0x00 /\* Online \*/
- #define [DMS\\_PM\\_LOW](#) 0x01 /\* Low Power \*/
- #define [DMS\\_PM\\_FACTORY](#) 0x02 /\* Factory Test Mode \*/
- #define [DMS\\_PM\\_OFFLINE](#) 0x03 /\* Offline \*/
- #define [DMS\\_PM\\_RESET](#) 0x04 /\* Reset \*/
- #define [DMS\\_PM\\_SHUT\\_DOWN](#) 0x05 /\* Shut Down \*/
- #define [DMS\\_PM\\_PERSISTENT\\_LOW](#) 0x06 /\* Persistent Low Power \*/
- #define [DMS\\_SET\\_REPORT\\_ENABLE](#) 1
- #define [DMS\\_SET\\_REPORT\\_DISABLE](#) 0
- #define [DMS\\_SWI\\_SET\\_IND\\_ENABLE](#) 1
- #define [DMS\\_SWI\\_SET\\_IND\\_DISABLE](#) 0

## Functions

- int [pack\\_dms\\_GetIMSI](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetIMSI](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetIMSI\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetModelID](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetModelID](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetModelID\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetFirmwareInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetFirmwareInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetFirmwareInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetPower](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetPower](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetPower\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetSerialNumbers](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetSerialNumbers](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetSerialNumbers\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetHardwareRevision](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetHardwareRevision](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetHardwareRevision\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSGetBandCapability](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_SLQSGetBandCapability](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSGetBandCapability\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetDeviceCapabilities](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)

- int [unpack\\_dms\\_GetDeviceCapabilities](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetDeviceCapabilities\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetFirmwareRevisions](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetFirmwareRevisions](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetFirmwareRevisions\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetFirmwareRevision](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetFirmwareRevision](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetFirmwareRevision\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetDeviceSerialNumbers](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetDeviceSerialNumbers](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetDeviceSerialNumbers\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetPRLVersion](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetPRLVersion](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetPRLVersion\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetNetworkTime](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetNetworkTime](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetNetworkTime\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetVoiceNumber](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetVoiceNumber](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetVoiceNumber\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetDeviceHardwareRev](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetDeviceHardwareRev](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetDeviceHardwareRev\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetFSN](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetFSN](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetFSN\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetDeviceCap](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetDeviceCap](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetDeviceCap\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetPower](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetPower\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetPower](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetPower\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetBandCapability](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetBandCapability](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetBandCapability\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetUSBComp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetUSBComp](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetUSBComp\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetUSBComp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetUSBComp\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetUSBComp](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetUSBComp\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetCustFeature](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetCustFeature\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetCustFeature](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetCustFeature\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetCustFeature](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetCustFeature](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetCustFeature\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetFirmwarePreference](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_SetFirmwarePreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetFirmwarePreference\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetCrashAction](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_GetCrashAction](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetCrashAction\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetCrashAction](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetCrashAction\\_t](#) reqArg)
- int [unpack\\_dms\\_SetCrashAction](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetCrashAction\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetDeviceMfr](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)

- int [unpack\\_dms\\_GetDeviceMfr](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetDeviceMfr\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetEventReport](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetEventReport\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetEventReport](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetEventReport\\_t](#) \*pOutput)
- int [unpack\\_dms\\_SetEventReport\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetEventReport\\_ind\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMGetICCID](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMGetICCID\\_t](#) \*reqArg)
- int [unpack\\_dms\\_UIMGetICCID](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMGetICCID\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetCustFeaturesV2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetCustFeaturesV2\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetCustFeaturesV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetCustFeaturesV2\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetCustFeaturesV2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_GetCustFeaturesV2\\_t](#) \*reqArg)
- int [unpack\\_dms\\_GetCustFeaturesV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetCustFeaturesV2\\_t](#) \*pOutput)
- int [pack\\_dms\\_GetActivationState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_GetActivationState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_GetActivationState\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetFirmwareCurr](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetFirmwareCurr](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetFirmwareCurr\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiSetDyingGaspCfg](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiSetDyingGaspCfg\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SLQSSwiSetDyingGaspCfg](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiSetDyingGaspCfg\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiClearDyingGaspStatistics](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiClearDyingGaspStatistics](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiClearDyingGaspStatistics\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetDyingGaspStatistics](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetDyingGaspStatistics](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetDyingGaspStatistics\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetDyingGaspCfg](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSDmsSwiGetResetInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSDmsSwiGetResetInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_t](#) \*pOutput)
- int [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_ind\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSDmsSwiIndicationRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSDmsSwiIndicationRegister\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SLQSDmsSwiIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSDmsSwiIndicationRegister\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetFwUpdateStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetFwUpdateStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetFwUpdateStatus\\_t](#) \*pOutput)

### 9.3.1 Macro Definition Documentation

9.3.1.1 `#define DMS_IMGDETAILS_LEN 16`

9.3.1.2 `#define DMS_MAX_CUST_ID_LEN 64`

```
9.3.1.3  #define DMS_MAX_CUST_VALUE_LEN 8

9.3.1.4  #define DMS_MAX_FWUPDATE_LOG_STR_SZ 255

9.3.1.5  #define DMS_MAX_FWUPDATE_REF_STR_SZ 15

9.3.1.6  #define DMS_PM_FACTORY 0x02 /* Factory Test Mode */

9.3.1.7  #define DMS_PM_LOW 0x01 /* Low Power */

9.3.1.8  #define DMS_PM_OFFLINE 0x03 /* Offline */

9.3.1.9  #define DMS_PM_ONLINE 0x00 /* Online */

9.3.1.10 #define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */

9.3.1.11 #define DMS_PM_RESET 0x04 /* Reset */

9.3.1.12 #define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */

9.3.1.13 #define DMS_SET_REPORT_DISABLE 0

9.3.1.14 #define DMS_SET_REPORT_ENABLE 1

9.3.1.15 #define DMS_SLQSFWINFO_APPVERSION_SZ 85

9.3.1.16 #define DMS_SLQSFWINFO_BOOTVERSION_SZ 85

9.3.1.17 #define DMS_SLQSFWINFO_CARRIER_SZ 20

9.3.1.18 #define DMS_SLQSFWINFO_CUR_CARR_NAME 17

9.3.1.19 #define DMS_SLQSFWINFO_CUR_CARR_REV 13

9.3.1.20 #define DMS_SLQSFWINFO_MODELID_SZ 20

9.3.1.21 #define DMS_SLQSFWINFO_PACKAGEID_SZ 85

9.3.1.22 #define DMS_SLQSFWINFO_PRIVERSION_SZ 16

9.3.1.23 #define DMS_SLQSFWINFO_SKU_SZ 15

9.3.1.24 #define DMS_SWI_SET_IND_DISABLE 0

9.3.1.25 #define DMS_SWI_SET_IND_ENABLE 1

9.3.1.26 #define DMS_UINT8_MAX_STRING_SZ 255

9.3.1.27 #define MAX_BUILD_ID_LEN 255

9.3.1.28 #define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160

9.3.1.29 #define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20

9.3.1.30 #define UNIQUE_ID_LEN 16
```

### 9.3.2 Function Documentation

#### 9.3.2.1 `int pack_dms_GetActivationState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get Activation State pack

##### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

##### Returns

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

##### See Also

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

#### 9.3.2.2 `int pack_dms_GetBandCapability ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Band Capability pack

##### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

##### Returns

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

##### See Also

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

#### 9.3.2.3 `int pack_dms_GetCrashAction ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Crash Action pack

##### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

##### Returns

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

## See Also

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

9.3.2.4 `int pack_dms_GetCustFeature ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use [pack\\_dms\\_GetCustFeatures-V2\(\)](#) instead for EM74xx/MC74xx.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.5 `int pack_dms_GetCustFeaturesV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_GetCustFeaturesV2_t * reqArg )`

9.3.2.6 `int pack_dms_GetDeviceCap ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Device Capabilities pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.7 `int pack_dms_GetDeviceCapabilities ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get device capability pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

### 9.3.2.8 int pack\_dms\_GetDeviceHardwareRev ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get Hardware Revision pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

### 9.3.2.9 int pack\_dms\_GetDeviceMfr ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get Manufacture pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

### 9.3.2.10 int pack\_dms\_GetDeviceSerialNumbers ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

get Device Serial Number pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.11** `int pack_dms_GetFirmwareInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get firmware info pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.12** `int pack_dms_GetFirmwareRevision ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get Firmware Revision pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.13** `int pack_dms_GetFirmwareRevisions ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get Firmware Revisions pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

**9.3.2.14** `int pack_dms_GetFSN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get FSN pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

**9.3.2.15** `int pack_dms_GetHardwareRevision ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get hardware revision pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

**9.3.2.16** `int pack_dms_GetIMSI ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get IMSI pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx-xx and all EM74xx firmware versions. Please use [pack\\_uim\\_ReadTransparent\(\)](#)(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.17** int pack\_dms\_GetModelID ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

get model id pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.18** int pack\_dms\_GetNetworkTime ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Get Network Time pack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

9.3.2.19 `int pack_dms_GetPower ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get power pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.3.2.20 `int pack_dms_GetPRLVersion ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get PRL Versions pack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.3.2.21 `int pack_dms_GetSerialNumbers ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get serial numbers pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

#### Returns

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

## See Also

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

9.3.2.22 `int pack_dms_GetUSBComp ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get USB Comp pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.23 `int pack_dms_GetVoiceNumber ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Voice Number pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.24 `int pack_dms_SetCrashAction ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetCrashAction_t reqArg )`

Set Crash Action pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

Set Crash Action unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

**9.3.2.25** int pack\_dms\_SetCustFeature ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_dms\_SetCustFeature\_t \* reqArg )

Set Custom Feature pack. This API is deprecated for EM74xx/MC74xx, please use [pack\\_dms\\_SetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

**9.3.2.26** int pack\_dms\_SetCustFeaturesV2 ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_dms\_SetCustFeaturesV2\_t \* reqArg )

Set Cust Features pack.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>pespLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.27** `int pack_dms_SetEventReport ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetEventReport_t * reqArg )`

Set Event Report pack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.28** `int pack_dms_SetFirmwarePreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Set Firmware Preference pack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

**Returns**

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

**See Also**

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

**9.3.2.29** `int pack_dms_SetPower ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetPower_t * reqArg )`

Set Power pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.30 int pack\_dms\_SetUSBComp ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_SetUSBComp\_t \* *reqArg* )

Set USB Comp pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.31 int pack\_dms\_SLQSDmsSwiGetResetInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

To get reset info pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.3.2.32** `int pack_dms_SLQSDmsSwiIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSDmsSwiIndicationRegister_t * reqArg )`

Set the registration state for different indication pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### Note

support EM/MC74xx onwards

#### See Also

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

**9.3.2.33** `int pack_dms_SLQSGetBandCapability ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

get band capability pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

**9.3.2.34** `int pack_dms_SLQSSwiClearDyingGaspStatistics ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Clear Dying GASP Statistics pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.3.2.35** `int pack_dms_SLQSSwiGetDyingGaspCfg ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get Dying GASP Config pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.3.2.36** `int pack_dms_SLQSSwiGetDyingGaspStatistics ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get Dying GASP Statistics pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.3.2.37** `int pack_dms_SLQSSwiGetFirmwareCurr ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get currently active image pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.3.2.38** `int pack_dms_SLQSSwiGetFwUpdateStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To get Firmware Update status pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.3.2.39** `int pack_dms_SLQSSwiSetDyingGaspCfg ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetDyingGaspCfg_t * reqArg )`

Set Dying GASP Config pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.3.2.40** `int pack_dms_UIMGetICCID ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMGetICCID_t * reqArg )`

Packs the UIMGetICCID response message to a user-provided response structure. This API is deprecated on MC73xx/EM73xx modules. Since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use [pack\\_uim\\_ReadTransparent\(\)](#)(EF ID: 3F00 2FE2) instead for new firmware versions and new modules.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.3.2.41 int unpack\_dms\_GetActivationState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetActivationState\_t \* *pOutput* )

Get Activation State unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.42 int unpack\_dms\_GetBandCapability ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetBandCapability\_t \* *pOutput* )

Get Band Capabilities unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.43 `int unpack_dms_GetCrashAction ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetCrashAction_t * pOutput )`

Get Crash Action unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.3.2.44 `int unpack_dms_GetCustFeature ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeature_t * pOutput )`

Get Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use [unpack\\_dms\\_GetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.3.2.45 `int unpack_dms_GetCustFeaturesV2 ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t * pOutput )`

9.3.2.46 `int unpack_dms_GetDeviceCap ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetDeviceCap_t * pOutput )`

Get Device Capabilities unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.47** int unpack\_dms\_GetDeviceCapabilities ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetDeviceCapabilities\_t \* *pOutput* )

get device capability unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.48** int unpack\_dms\_GetDeviceHardwareRev ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetDeviceHardwareRev\_t \* *pOutput* )

Get Hardware Revision unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.49** int unpack\_dms\_GetDeviceMfr ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetDeviceMfr\_t \* *pOutput* )

Get Manufacture unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.50** int unpack\_dms\_GetDeviceSerialNumbers ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_GetDeviceSerialNumbers\_t \* *pOutput* )

get Device Serial Number unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.51** int unpack\_dms\_GetFirmwareInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetFirmwareInfo\_t \*  
*pOutput* )

get firmware info unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.52** int unpack\_dms\_GetFirmwareRevision ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetFirmware-  
Revision\_t \* *pOutput* )

get Firmware Revision unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.53** `int unpack_dms_GetFirmwareRevisions ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetFirmwareRevisions_t * pOutput )`

get Firmware Revisions unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.54** `int unpack_dms_GetFSN ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetFSN_t * pOutput )`

Get FSN unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.55** `int unpack_dms_GetHardwareRevision ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetHardwareRevision_t * pOutput )`

get hardware revision unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

### 9.3.2.56 int unpack\_dms\_GetIMSI ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetIMSI\_t \* *pOutput* )

get IMSI unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx-xx\_xx and all EM74xx firmware versions. Please use [unpack\\_uim\\_ReadTransparent\(\)](#) (EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

### 9.3.2.57 int unpack\_dms\_GetModelID ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetModelID\_t \* *pOutput* )

get model id unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

### 9.3.2.58 int unpack\_dms\_GetNetworkTime ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetNetworkTime\_t \* *pOutput* )

Get Network Time unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.59 int unpack\_dms\_GetPower ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetPower\_t \* *pOutput* )

get power unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.60 int unpack\_dms\_GetPRLVersion ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetPRLVersion\_t \* *pOutput* )

Get PRL Versions unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.61 int unpack\_dms\_GetSerialNumbers ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetSerialNumbers\_t \* *pOutput* )

get serial numbers unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.62** `int unpack_dms_GetUSBComp ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetUSBComp_t * pOutput )`

Get USB Comp unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.63** `int unpack_dms_GetVoiceNumber ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t * pOutput )`

Get Voice Number unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.64** `int unpack_dms_SetCrashAction ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetCrashAction_t * pOutput )`

Set Crash Action unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response. Not used

## Returns

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

## See Also

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

**9.3.2.65** `int unpack_dms_SetCustFeature ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetCustFeature_t * pOutput )`

Set Custom Feature unpack. This API is deprecated for EM74xx/MC74xx, please use [unpack\\_dms\\_SetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.66** `int unpack_dms_SetCustFeaturesV2 ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetCustFeaturesV2_t * pOutput )`

Set Cust features unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.67** `int unpack_dms_SetEventReport ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetEventReport_t * pOutput )`

Set Event Report unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.68 int unpack\_dms\_SetEventReport\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetEventReport\_ind\_t \* *pOutput* )

Event Report Indication unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.69 int unpack\_dms\_SetFirmwarePreference ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetFirmwarePreference\_t \* *pOutput* )

Set Firmware Preference unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.70 int unpack\_dms\_SetPower ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetPower\_t \* *pOutput* )

Set Power unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.71 int unpack\_dms\_SetUSBComp ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetUSBComp\_t \* *pOutput* )

Set USB Comp unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.72 int unpack\_dms\_SLQSDmsSwiGetResetInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSDmsSwiGetResetInfo\_t \* *pOutput* )

To get reset info unpack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

9.3.2.73 int unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t \* *pOutput* )

DMS reset info Indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**Note**

support EM/MC74xx onwards

**See Also**

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

**9.3.2.74** int unpack\_dms\_SLQSDmsSwilIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSDmsSwilIndicationRegister\_t \* *pOutput* )

Set the registration state for different indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**Note**

support EM/MC74xx onwards

**See Also**

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

**9.3.2.75** int unpack\_dms\_SLQSGetBandCapability ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSGetBand-  
Capability\_t \* *pOutput* )

get band capability unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.76** `int unpack_dms_SLQSSwiClearDyingGaspStatistics ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiClearDyingGaspStatistics_t * pOutput )`

Clear Dying GASP Statistics unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.3.2.77** `int unpack_dms_SLQSSwiGetDyingGaspCfg ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspCfg_t * pOutput )`

Get Dying GASP Config unpack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.3.2.78** `int unpack_dms_SLQSSwiGetDyingGaspStatistics ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSwiGetDyingGaspStatistics_t * pOutput )`

Get Dying GASP Statistics unpack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.3.2.79** int unpack\_dms\_SLQSSwiGetFirmwareCurr ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetFirmwareCurr\_t \* *pOutput* )

get currently active image unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.80** int unpack\_dms\_SLQSSwiGetFwUpdateStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t \* *pOutput* )

To get Firmware Update status unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.3.2.81** int unpack\_dms\_SLQSSwiSetDyingGaspCfg ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t \* *pOutput* )

Set Dying GASP Config unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.3.2.82 int unpack\_dms\_UIMGetICCID ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMGetICCID\_t \* *pOutput* )

Unpacks the UIMGetICCID response message to a user-provided response structure.

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

## 9.4 fms.h File Reference

### Data Structures

- struct [CarrierImage\\_t](#)
- struct [pack\\_fms\\_GetImagesPreference\\_t](#)
- struct [FMSImageElement](#)
- struct [FMSPrefImageList](#)
- struct [unpack\\_fms\\_GetImagesPreference\\_t](#)
- struct [pack\\_fms\\_GetStoredImages\\_t](#)
- struct [FMSImageIdElement](#)
- struct [FMSImageIdEntries](#)
- struct [FMSImageList](#)
- struct [unpack\\_fms\\_GetStoredImages\\_t](#)
- struct [pack\\_fms\\_SetImagesPreference\\_t](#)
- struct [unpack\\_fms\\_SetImagesPreference\\_t](#)

### Macros

- #define [FMS\\_GOBI\\_MBN\\_IMG\\_ID\\_STR\\_LEN](#) 16
- #define [FMS\\_GOBI\\_MBN\\_BUILD\\_ID\\_STR\\_LEN](#) 100
- #define [FMS\\_GOBI\\_LISTENTRIES\\_MAX](#) 2
- #define [FMS\\_MAX\\_IMAGE\\_PREFERENCE\\_IMAGE\\_SIZE](#) 255
- #define [FMS\\_MAX\\_IMAGE\\_ID\\_ELEMENT](#) 50
- #define [FMS\\_IMAGE\\_ID\\_MAX\\_ENTRIES](#) 2
- #define [FMS\\_FW\\_PRI\\_BUILD\\_MATCH\\_LEN](#) 11
- #define [FMS\\_IMAGE\\_ID\\_IMG\\_ID\\_LEN](#) 16
- #define [FMS\\_IMAGE\\_ID\\_BUILD\\_ID\\_LEN](#) 32
- #define [FMS\\_IMAGE\\_ID\\_PRI\\_IMGTYPE](#) 0x01

## Functions

- int [pack\\_fms\\_GetImagesPreference](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_fms\\_GetImagesPreference\\_t](#) \*reqArg)
- int [unpack\\_fms\\_GetImagesPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_fms\\_GetImagesPreference\\_t](#) \*pOutput)
- int [pack\\_fms\\_GetStoredImages](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_fms\\_GetStoredImages\\_t](#) \*reqArg)
- int [unpack\\_fms\\_GetStoredImages](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_fms\\_GetStoredImages\\_t](#) \*pOutput)
- int [pack\\_fms\\_SetImagesPreference](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_fms\\_SetImagesPreference\\_t](#) \*reqArg)
- int [unpack\\_fms\\_SetImagesPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_fms\\_SetImagesPreference\\_t](#) \*pOutput)
- uint32\_t [GetValidFwPriCombinations](#) ([FMSImageList](#) \*pStoredImageList, uint32\_t \*pValidCombinationSize, [CarrierImage\\_t](#) \*pValidCombinations)

### 9.4.1 Macro Definition Documentation

9.4.1.1 `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`

9.4.1.2 `#define FMS_GOBI_LISTENTRIES_MAX 2`

9.4.1.3 `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN 100`

9.4.1.4 `#define FMS_GOBI_MBN_IMG_ID_STR_LEN 16`

9.4.1.5 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.4.1.6 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.4.1.7 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.4.1.8 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.4.1.9 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.4.1.10 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

### 9.4.2 Function Documentation

9.4.2.1 `uint32_t GetValidFwPriCombinations ( FMSImageList * pStoredImageList, uint32_t * pValidCombinationSize, CarrierImage\_t * pValidCombinations )`

This API distills valid Firmware/PRI combinations from `GetStoredImages` result

#### Parameters

in	<i>pStoredImageList</i>	<ul style="list-style-type: none"> <li>• image list returned from <code>GetStoredImages</code></li> <li>• See <a href="#">FMSImageList</a></li> </ul>
in, out	<i>pValidCombinationSize</i>	<ul style="list-style-type: none"> <li>• number of combination passed in and returned</li> </ul>

out	<i>pValid-Combinations</i>	<ul style="list-style-type: none"> <li>• valid combinations returned</li> <li>• See <a href="#">CarrierImage_t</a></li> </ul>
-----	----------------------------	---

## Returns

- eQCWWAN\_ERR\_INVALID\_ARG - Invalid parameters
- eQCWWAN\_ERR\_BUFFER\_SZ - No enough element to store combinatons returned

## See Also

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

9.4.2.2 `int pack_fms_GetImagesPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_GetImagesPreference_t * reqArg )`

Get Images Preference pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.3 `int pack_fms_GetStoredImages ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_GetStoredImages_t * reqArg )`

Get Images Preference pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.4 `int pack_fms_SetImagesPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_fms_SetImagesPreference_t * reqArg )`

Set Images Preference pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

9.4.2.5 `int unpack_fms_GetImagesPreference ( uint8_t * pResp, uint16_t respLen, unpack_fms_GetImagesPreference_t * pOutput )`

Get Images Preference unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

9.4.2.6 `int unpack_fms_GetStoredImages ( uint8_t * pResp, uint16_t respLen, unpack_fms_GetStoredImages_t * pOutput )`

Get Images Preference unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

9.4.2.7 `int unpack_fms_SetImagesPreference ( uint8_t * pResp, uint16_t respLen, unpack_fms_SetImagesPreference_t * pOutput )`

Set Images Preference unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## 9.5 loc.h File Reference

### Data Structures

- struct [loc\\_LocApplicationInfo](#)
- struct [loc\\_SV](#)
- struct [loc\\_SVInfo](#)
- struct [loc\\_GnssData](#)
- struct [loc\\_CellDb](#)
- struct [loc\\_ClkInfo](#)
- struct [loc\\_BdsSV](#)
- struct [loc\\_BdsSVInfo](#)
- struct [pack\\_loc\\_EventRegister\\_t](#)
- struct [unpack\\_loc\\_EventRegister\\_t](#)
- struct [pack\\_loc\\_SetExtPowerState\\_t](#)
- struct [unpack\\_loc\\_SetExtPowerState\\_t](#)
- struct [pack\\_loc\\_Start\\_t](#)
- struct [unpack\\_loc\\_Start\\_t](#)
- struct [pack\\_loc\\_Stop\\_t](#)
- struct [unpack\\_loc\\_Stop\\_t](#)

- struct [pack\\_loc\\_SetOperationMode\\_t](#)
- struct [unpack\\_loc\\_SetOperationMode\\_t](#)
- struct [pack\\_loc\\_Delete\\_Assist\\_Data\\_t](#)
- struct [unpack\\_loc\\_Delete\\_Assist\\_Data\\_t](#)
- struct [loc\\_precisionDilution](#)
- struct [loc\\_sensorDataUsage](#)
- struct [loc\\_svUsedforFix](#)
- struct [loc\\_gpsTime](#)
- struct [unpack\\_loc\\_PositionRpt\\_Ind\\_t](#)
- struct [unpack\\_loc\\_EngineState\\_Ind\\_t](#)
- struct [unpack\\_loc\\_SetExtPowerConfig\\_Ind\\_t](#)
- struct [unpack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#)
- struct [pack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#)
- struct [unpack\\_loc\\_BestAvailPos\\_Ind\\_t](#)

## Macros

- `#define LOC_UINT8_MAX_STRING_SZ 255`
- `#define LOCEVENTMASKPOSITIONREPORT 0x00000001`
- `#define LOCEVENTMASKGNSSSVINFO 0x00000002`
- `#define LOCEVENTMASKNMEA 0x00000004`
- `#define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008`
- `#define LOCEVENTMASKINJECTTIMERREQ 0x00000010`
- `#define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020`
- `#define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040`
- `#define LOCEVENTMASKENGINESTATE 0x00000080`
- `#define LOCEVENTMASKFIXSESSIONSTATE 0x00000100`
- `#define LOCEVENTMASKWIFIREQ 0x00000200`
- `#define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400`
- `#define LOCEVENTMASKTIMESYNCREQ 0x00000800`
- `#define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000`
- `#define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000`
- `#define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000`
- `#define LOCEVENTMASKGEOFENCEGENALERT 0x00008000`
- `#define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000`
- `#define LOCEVENTMASKPEDOMETERCONTROL 0x00020000`
- `#define LOCEVENTMASKMOTIONDATACONTROL 0x00040000`
- `#define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000`
- `#define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000`
- `#define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000`
- `#define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000`
- `#define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000`
- `#define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000`
- `#define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF`

## Enumerations

- enum {  
[eQMI\\_LOC\\_SESS\\_STATUS\\_SUCCESS](#) =0,  
[eQMI\\_LOC\\_SESS\\_STATUS\\_IN\\_PROGRESS](#) =1,  
[eQMI\\_LOC\\_SESS\\_STATUS\\_FAILURE](#) =2,  
[eQMI\\_LOC\\_SESS\\_STATUS\\_TIMEOUT](#) =3 }

## Functions

- int [pack\\_loc\\_EventRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_EventRegister\\_t](#) \*reqArg)
- int [unpack\\_loc\\_EventRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_EventRegister\\_t](#) \*pOutput)
- int [pack\\_loc\\_SetExtPowerState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SetExtPowerState\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SetExtPowerState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetExtPowerState\\_t](#) \*pOutput)
- int [pack\\_loc\\_Start](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_Start\\_t](#) \*reqArg)
- int [unpack\\_loc\\_Start](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_Start\\_t](#) \*pOutput)
- int [pack\\_loc\\_Stop](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_Stop\\_t](#) \*reqArg)
- int [unpack\\_loc\\_Stop](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_Stop\\_t](#) \*pOutput)
- int [pack\\_loc\\_SetOperationMode](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SetOperationMode\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SetOperationMode](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetOperationMode\\_t](#) \*pOutput)
- int [pack\\_loc\\_DeleteAssistData](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_Delete\\_Assist\\_Data\\_t](#) \*reqArg)
- int [unpack\\_loc\\_DeleteAssistData](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_Delete\\_Assist\\_Data\\_t](#) \*pOutput)
- int [unpack\\_loc\\_PositionRpt\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_PositionRpt\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_EngineState\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_EngineState\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_SetExtPowerConfig\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetExtPowerConfig\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCGetBestAvailPos](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCGetBestAvailPos](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCGetBestAvailPos\\_t](#) \*pOutput)
- int [unpack\\_loc\\_BestAvailPos\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_BestAvailPos\\_Ind\\_t](#) \*pOutput)

### 9.5.1 Macro Definition Documentation

9.5.1.1 `#define LOC_UINT8_MAX_STRING_SZ 255`

9.5.1.2 `#define LOCEVENTMASKBATCHFULLNOTIFICATION 0x00080000`

The control point must enable this mask to receive notification when a batch is full. The location engine sends this event to notify of Batch Full for ongoing batching session.

9.5.1.3 `#define LOCEVENTMASKENGINESTATE 0x00000080`

The control point must enable this mask to receive engine state report event indications.

9.5.1.4 `#define LOCEVENTMASKFIXSESSIONSTATE 0x00000100`

The control point must enable this mask to receive fix session status report event indications.

9.5.1.5 `#define LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION 0x00400000`

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach notification is for multiple Geofences. Breaches from multiple Geofences are all batched and sent in the same notification.

**9.5.1.6 #define LOCEVENTMASKGEOFENCEBREACHNOTIFICATION 0x00010000**

The control point must enable this mask to receive notifications when a Geofence is breached. These events are generated when a UE enters or leaves the perimeter of a Geofence. This breach report is for a single Geofence.

**9.5.1.7 #define LOCEVENTMASKGEOFENCEGENALERT 0x00008000**

The control point must enable this mask to receive Geofence alerts. These alerts are generated to inform the client of the changes that may affect a Geofence, for example, if GPS is turned off or if the network is unavailable.

**9.5.1.8 #define LOCEVENTMASKGNSSMEASUREMENTREPORT 0x01000000**

The control point must enable this mask to receive system clock and satellite measurement report events (system clock, [SV](#) time, Doppler, etc.). Reports are generated only for the GNSS satellite constellations that are enabled using QMI\_LOC\_SET\_GNSS\_CONSTELL\_REPORT\_CONFIG(Not yet supported).

**9.5.1.9 #define LOCEVENTMASKGNSSSVINFO 0x00000002**

The control point must enable this mask to receive satellite report event indications. These reports are sent at a 1 Hz rate.

**9.5.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040**

The control point must enable this mask to receive position injection request event indications.

**9.5.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020**

The control point must enable this mask to receive predicted orbits request event indications.

**9.5.1.12 #define LOCEVENTMASKINJECTTIMERREQ 0x00000010**

The control point must enable this mask to receive time injection request event indications.

**9.5.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000**

The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications.

**9.5.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF**

Invalid Event Mask

**9.5.1.15 #define LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT 0x00100000**

The control point must enable this mask to receive position report indications along with an ongoing batching session. The location engine sends this event to notify the batched position report while a batching session is ongoing.

**9.5.1.16 #define LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ 0x00002000**

The control point must enable this mask to receive location server requests. These requests are generated when the service wishes to establish a connection with a location server.

**9.5.1.17 #define LOCEVENTMASKMOTIONDATACONTROL 0x00040000**

The control point must enable this mask to register for motion data control requests from the location engine. The location engine sends this event to control the injection of motion data.

**9.5.1.18 #define LOCEVENTMASKNIGEOFENCENOTIFICATION 0x00004000**

The control point must enable this mask to receive notifications related to network-initiated Geofences. These events notify the client when a network-initiated Geofence is added, deleted, or edited.

**9.5.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008**

The control point must enable this mask to receive NI Notify/Verify request event indications.

**9.5.1.20 #define LOCEVENTMASKNMEA 0x00000004**

The control point must enable this mask to receive NMEA reports for position and satellites in view. The report is at a 1 Hz rate.

**9.5.1.21 #define LOCEVENTMASKPEDOMETERCONTROL 0x00020000**

The control point must enable this mask to register for pedometer control requests from the location engine. The location engine sends this event to control the injection of pedometer reports.

**9.5.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001**

The control point must enable this mask to receive position report event indications.

**9.5.1.23 #define LOCEVENTMASKSENSORSTREAMINGREADYSTATUS 0x00000400**

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept data from the sensors (accelerometer, gyroscope, etc.).

**9.5.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000**

The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications.

**9.5.1.25 #define LOCEVENTMASKTIMESYNCREQ 0x00000800**

The control point must enable this mask to receive time sync requests from the GPS engine. Time sync enables the GPS engine to synchronize its clock with the sensor processor's clock.

**9.5.1.26 #define LOCEVENTMASKVEHICLEDATAREADYSTATUS 0x00800000**

The control point must enable this mask to receive notifications from the location engine indicating its readiness to accept vehicle data (vehicle accelerometer, vehicle angular rate, vehicle odometry, etc.).

**9.5.1.27 #define LOCEVENTMASKWIFIREQ 0x00000200**

The control point must enable this mask to receive Wi-Fi position request event indications.

## 9.5.2 Enumeration Type Documentation

### 9.5.2.1 anonymous enum

Enumerator

***eQMI\_LOC\_SESS\_STATUS\_SUCCESS***  
***eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS***  
***eQMI\_LOC\_SESS\_STATUS\_FAILURE***  
***eQMI\_LOC\_SESS\_STATUS\_TIMEOUT***

## 9.5.3 Function Documentation

**9.5.3.1** `int pack_loc_DeleteAssistData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Delete_Assist_Data_t * reqArg )`

Delete Assistant Data pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

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

See Also

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

**9.5.3.2** `int pack_loc_EventRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_EventRegister_t * reqArg )`

Event Register pack

Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

Returns

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

See Also

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

**9.5.3.3** `int pack_loc_SetExtPowerState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SetExtPowerState_t * reqArg )`

Set Ext Power State pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.4 int pack\_loc\_SetOperationMode ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_loc\_SetOperationMode\_t \* reqArg )

Set Operation Mode pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.5 int pack\_loc\_SLQSLOCGetBestAvailPos ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_loc\_SLQSLOCGetBestAvailPos\_t \* reqArg )

Get Best Avail position pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.6 int pack\_loc\_Start ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_loc\_Start\_t \* reqArg )

LOC Start pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.7 int pack\_loc\_Stop ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_loc\_Stop\_t \* reqArg )

Loc Stop pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.8 int unpack\_loc\_BestAvailPos\_Ind ( uint8\_t \* pResp, uint16\_t respLen, unpack\_loc\_BestAvailPos\_Ind\_t \* pOutput )

Loc Best Avail position Indication unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.9 int unpack\_loc\_DeleteAssistData ( uint8\_t \* pResp, uint16\_t respLen, unpack\_loc\_Delete\_Assist\_Data\_t \* pOutput )

Delete Assistant Data unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

9.5.3.10 int unpack\_loc\_EngineState\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_EngineState\_Ind\_t \* *pOutput* )

Loc Engine State Indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

9.5.3.11 int unpack\_loc\_EventRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_EventRegister\_t \* *pOutput* )

Event Register unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

9.5.3.12 int unpack\_loc\_PositionRpt\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_PositionRpt\_Ind\_t \* *pOutput* )

Loc Position Report Indication unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.13 int unpack\_loc\_SetExtPowerConfig\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SetExtPowerConfig\_Ind\_t \* *pOutput* )

Loc Set External Power Configure Indication unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.14 int unpack\_loc\_SetExtPowerState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SetExtPowerState\_t \* *pOutput* )

Set Ext Power State unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.5.3.15 `int unpack_loc_SetOperationMode ( uint8_t * pResp, uint16_t respLen, unpack_loc_SetOperationMode_t * pOutput )`

Set Operation Mode unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.5.3.16 `int unpack_loc_SLQSLOCGetBestAvailPos ( uint8_t * pResp, uint16_t respLen, unpack_loc_SLQSLOCGetBestAvailPos_t * pOutput )`

Get Best Avail position unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.5.3.17 `int unpack_loc_Start ( uint8_t * pResp, uint16_t respLen, unpack_loc_Start_t * pOutput )`

Loc Start unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.5.3.18 int unpack\_loc\_Stop ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_Stop\_t \* *pOutput* )

Loc Stop unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

## 9.6 nas.h File Reference

### Data Structures

- struct [unpack\\_nas\\_GetSignalStrengths\\_t](#)
- struct [unpack\\_nas\\_SLQSGetSysSelectionPref\\_t](#)
- struct [nas\\_netSelectionPref](#)
- struct [nas\\_acqOrderPref](#)
- struct [nas\\_CSGID](#)
- struct [pack\\_nas\\_SLQSSetSysSelectionPref\\_t](#)
- struct [pack\\_nas\\_SLQSNasIndicationRegisterExt\\_t](#)
- struct [RFBandInfoElements](#)
- struct [unpack\\_nas\\_GetRFInfo\\_t](#)
- struct [cdmaSSInfo](#)
- struct [hdrSSInfo](#)
- struct [lteSSInfo](#)
- struct [tdscdmaSigInfoExt](#)
- struct [unpack\\_nas\\_SLQSNasGetSigInfo\\_t](#)
- struct [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t](#)
- struct [unpack\\_nas\\_GetHomeNetwork\\_t](#)
- struct [nas\\_SrvStatusInfo](#)
- struct [nas\\_GSMSrvStatusInfo](#)
- struct [nas\\_sysInfoCommon](#)
- struct [nas\\_CDMASysInfo](#)
- struct [nas\\_HDRSysInfo](#)
- struct [nas\\_GSMSysInfo](#)
- struct [nas\\_WCDMASysInfo](#)
- struct [nas\\_LTESysInfo](#)
- struct [nas\\_AddCDMASysInfo](#)
- struct [nas\\_AddSysInfo](#)
- struct [nas\\_CallBarringSysInfo](#)
- struct [unpack\\_nas\\_SLQSGetSysInfo\\_t](#)
- struct [unpack\\_nas\\_SLQSSysInfoCallback\\_ind\\_t](#)
- struct [unpack\\_nas\\_GetServingNetwork\\_t](#)
- struct [unpack\\_nas\\_GetServingNetworkCapabilities\\_t](#)
- struct [nas\\_QmiNas3GppNetworkInfo](#)

- struct [nas\\_QmiNas3GppNetworkRAT](#)
- struct [nas\\_QmisNasPcsDigit](#)
- struct [unpack\\_nas\\_PerformNetworkScan\\_t](#)
- struct [unpack\\_nas\\_SLQSSwiGetLteCQI\\_t](#)
- struct [nas\\_CommInfo](#)
- struct [nas\\_LTEInfo](#)
- struct [unpack\\_nas\\_SLQSNasSwiModemStatus\\_t](#)
- struct [nas\\_servSystem](#)
- struct [nas\\_dataSrvCapabilities](#)
- struct [nas\\_currentPLMN](#)
- struct [nas\\_roamIndList](#)
- struct [nas\\_qaQmi3Gpp2TimeZone](#)
- struct [nas\\_detailSvcInfo](#)
- struct [nas\\_CDMASysInfoExt](#)
- struct [nas\\_callBarStatus](#)
- struct [unpack\\_nas\\_SLQSGetServingSystem\\_t](#)
- struct [nas\\_rxSignalStrengthListElement](#)
- struct [nas\\_ecioListElement](#)
- struct [nas\\_errorRateListElement](#)
- struct [nas\\_rsrqInformation](#)
- struct [nas\\_lteSnrInformation](#)
- struct [nas\\_lteRsrpInformation](#)
- struct [unpack\\_nas\\_SLQSGetSignalStrength\\_t](#)
- struct [nas\\_SLQSSignalStrengthsIndReq](#)
- struct [pack\\_nas\\_SLQSSetSignalStrengthsCallback\\_t](#)
- struct [nas\\_SLQSSignalStrengthsInformation](#)
- struct [nas\\_RejectReasonTlv](#)
- struct [nas\\_SignalStrengthTlv](#)
- struct [nas\\_RFInfoTlv](#)
- struct [nas\\_SLQSSignalStrengthsTlv](#)
- struct [unpack\\_nas\\_SetEventReportInd\\_t](#)
- struct [unpack\\_nas\\_GetCDMANetworkParameters\\_t](#)
- struct [pack\\_nas\\_SetACCOLC\\_t](#)
- struct [nas\\_CDMARSSIThresh](#)
- struct [nas\\_CDMAECIOThresh](#)
- struct [nas\\_HDRRSSIThresh](#)
- struct [nas\\_HDRECIOThresh](#)
- struct [nas\\_HDRSINRThreshold](#)
- struct [nas\\_HDRIOThresh](#)
- struct [nas\\_GSMRSSIThresh](#)
- struct [nas\\_WCDMARSSIThresh](#)
- struct [nas\\_WCDMAECIOThresh](#)
- struct [nas\\_LTERSSIThresh](#)
- struct [nas\\_LTESNRThreshold](#)
- struct [nas\\_LTERSRQThresh](#)
- struct [nas\\_LTERSRPThresh](#)
- struct [nas\\_LTESigRptConfig](#)
- struct [nas\\_TDSCDMARSCPTthresh](#)
- struct [nas\\_TDSCDMARSSIThresh](#)
- struct [nas\\_TDSCDMAECIOThresh](#)
- struct [nas\\_TDSCDMASINRThresh](#)
- struct [pack\\_nas\\_SLQSNasConfigSigInfo2\\_t](#)
- struct [unpack\\_nas\\_SetDataCapabilitiesCallback\\_ind\\_t](#)
- struct [unpack\\_nas\\_GetNetworkPreference\\_t](#)
- struct [pack\\_nas\\_SetNetworkPreference\\_t](#)

- struct [unpack\\_nas\\_SetNetworkPreference\\_t](#)
- struct [unpack\\_nas\\_SetRoamingIndicatorCallback\\_ind\\_t](#)
- struct [NAServingSystemInfo](#)
- struct [unpack\\_nas\\_SetServingSystemCallback\\_ind\\_t](#)
- struct [NASPhyCaAggScellIndType](#)
- struct [NASPhyCaAggScellIDBw](#)
- struct [NASPhyCaAggScellInfo](#)
- struct [NASPhyCaAggPcellInfo](#)
- struct [NASPhyCaAggScellIndex](#)
- struct [NasGetLTECphyCAInfo](#)
- struct [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t](#)
- struct [NASEmergencyModeTlv](#)
- struct [NASModePreferenceTlv](#)
- struct [NASBandPreferenceTlv](#)
- struct [NASPRLPreferenceTlv](#)
- struct [NASRoamPreferenceTlv](#)
- struct [NASLTEBandPreferenceTlv](#)
- struct [NASNetSelPreferenceTlv](#)
- struct [NASServDomainPrefTlv](#)
- struct [NASGWAcqOrderPrefTlv](#)
- struct [NASQmiCbkNasSystemSelPrefInd](#)
- struct [unpack\\_nas\\_SLQSSetSysSelectionPrefCallBack\\_ind\\_t](#)
- struct [NASOTAMessageTlv](#)
- struct [NASLteNasReleaseInfoTlv](#)
- struct [NASTimeInfoTlv](#)
- struct [NASQmiCbkNasSwiOTAMessageInd](#)
- struct [unpack\\_nas\\_SLQSNasSwiOTAMessageCallback\\_ind\\_t](#)
- struct [nas\\_MNRInfo](#)
- struct [pack\\_nas\\_SLQSInitiateNetworkRegistration\\_t](#)
- struct [pack\\_nas\\_SLQSNasSwiOTAMessageCallback\\_t](#)
- struct [pack\\_nas\\_SLQSGetPLMNName\\_t](#)
- struct [unpack\\_nas\\_SLQSGetPLMNName\\_t](#)
- struct [nas\\_nmrCellInfo](#)
- struct [nas\\_GERANInfo](#)
- struct [nas\\_geranInstInfo](#)
- struct [nas\\_UMTSinstInfo](#)
- struct [nas\\_UMTSInfo](#)
- struct [nas\\_CDMAInfo](#)
- struct [nas\\_cellParams](#)
- struct [nas\\_LTEInfoIntrafreq](#)
- struct [nas\\_infoInterFreq](#)
- struct [nas\\_LTEInfoInterfreq](#)
- struct [nas\\_gsmCellInfo](#)
- struct [nas\\_lteGsmCellInfo](#)
- struct [nas\\_LTEInfoNeighboringGSM](#)
- struct [nas\\_wcdmaCellInfo](#)
- struct [nas\\_lteWcdmaCellInfo](#)
- struct [nas\\_LTEInfoNeighboringWCDMA](#)
- struct [nas\\_umtsLTENbrCell](#)
- struct [nas\\_WCDMAInfoLTENeighborCell](#)
- struct [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t](#)
- struct [nas\\_timeInfo](#)
- struct [unpack\\_nas\\_SLQSGetNetworkTime\\_t](#)
- struct [nas\\_UniversalTime](#)
- struct [unpack\\_nas\\_SLQSNasNetworkTimeCallBack\\_ind\\_t](#)

- struct [nas\\_PhyCaAggScellIndType](#)
- struct [nas\\_PhyCaAggScellDIBw](#)
- struct [nas\\_PhyCaAggScellInfo](#)
- struct [nas\\_PhyCaAggPcellInfo](#)
- struct [nas\\_PhyCaAggScellIndex](#)
- struct [unpack\\_nas\\_SetNasLTECphyCaIndCallback\\_ind\\_t](#)

## Macros

- #define [NAS\\_OTA\\_MESSAGE\\_MAX\\_BUF\\_SIZE](#) 2048
- #define [NAS\\_MAX\\_NUM\\_NETWORKS](#) 30
- #define [NAS\\_MAX\\_DESCRIPTION\\_LENGTH](#) 255
- #define [NAS\\_PLMN\\_LENGTH](#) 3
- #define [NAS\\_SERVING\\_SYSTEM\\_INFO\\_MAX\\_RADIO\\_INTERFACE\\_LIST](#) 255

## Enumerations

- enum [LIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) {  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_DECONFIGURED](#) =0x00,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_CONFIGURED\\_DEACTIVATED](#) =0x01,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_CONFIGURED\\_ACTIVATED](#) =0x02 }
- enum [LIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) {  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_6](#) =0x00,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_15](#) =0x01,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_25](#) =0x02,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_50](#) =0x03,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_75](#) =0x04,  
[eLIBPACK\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_100](#) =0x05 }
- enum [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) {  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_6](#) =0x00,  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_15](#) =0x01,  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_25](#) =0x02,  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_50](#) =0x03,  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_75](#) =0x04,  
[eNAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE\\_100](#) =0x05 }
- enum [NAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_LITE](#) {  
[eNAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_DECONFIGURED\\_LITE](#) =0x00,  
[eNAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_CONFIGURED\\_DEACTIVATED\\_LITE](#) =0x01,  
[eNAS\\_LTE\\_CPHY\\_SCELL\\_STATE\\_CONFIGURED\\_ACTIVATED\\_LITE](#) =0x02 }

## Functions

- int [unpack\\_nas\\_GetSignalStrengths](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetSignalStrengths\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetSignalStrengths](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReq, uint16\_t \*pLen)
- int [pack\\_nas\\_SLQSGetSysSelectionPref](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSGetSysSelectionPref](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetSysSelectionPref\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSetSysSelectionPref](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSSetSysSelectionPref\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSSetSysSelectionPref](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SLQSSetBandPreference](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint64\_t bandPref)
- int [unpack\\_nas\\_SLQSSetBandPreference](#) (uint8\_t \*pResp, uint16\_t respLen)

- int [pack\\_nas\\_SLQSNasIndicationRegisterExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasIndicationRegisterExt\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasIndicationRegisterExt](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_GetRFInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetRFInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetRFInfo\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasGetSigInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNasGetSigInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetSigInfo\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_GetHomeNetwork](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetHomeNetwork\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetHomeNetwork](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [pack\\_nas\\_SLQSGetSysInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSGetSysInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetSysInfo\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasSysInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSysInfoCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetServingNetwork](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetServingNetwork](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetServingNetwork\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetServingNetworkCapabilities](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetServingNetworkCapabilities](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetServingNetworkCapabilities\\_t](#) \*pOutput)
- int [pack\\_nas\\_PerformNetworkScan](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_PerformNetworkScan](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_PerformNetworkScan\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSwiGetLteCQI](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSSwiGetLteCQI](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetLteCQI\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasSwiModemStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNasSwiModemStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasSwiModemStatus\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetServingSystem](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSGetServingSystem](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetServingSystem\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetSignalStrength](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint16\_t reqMask)
- int [unpack\\_nas\\_SLQSGetSignalStrength](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetSignalStrength\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSetSignalStrengthsCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSSetSignalStrengthsCallback\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSSetSignalStrengthsCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SetRFInfoCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pBenable)
- int [unpack\\_nas\\_SetRFInfoCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SetLURejectCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pBenable)
- int [unpack\\_nas\\_SetLURejectCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_nas\\_SetEventReportInd](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetEventReportInd\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetCDMANetworkParameters](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetCDMANetworkParameters](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetCDMA-NetworkParameters\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetANAAAAuthenticationStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetANAAAAuthenticationStatus](#) (uint8\_t \*pResp, uint16\_t respLen, uint32\_t \*pAuthStatus)
- int [pack\\_nas\\_GetACCOLC](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)

- int [unpack\\_nas\\_GetACCOLC](#) (uint8\_t \*pResp, uint16\_t respLen, uint8\_t \*pAccolc)
- int [pack\\_nas\\_SetACCOLC](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SetACCOLC\\_t](#) reqParam)
- int [unpack\\_nas\\_SetACCOLC](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SLQSNasConfigSigInfo2](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasConfigSigInfo2\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasConfigSigInfo2](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_nas\\_SetDataCapabilitiesCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetDataCapabilitiesCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetNetworkPreference](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetNetworkPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetNetworkPreference\\_t](#) \*pOutput)
- int [pack\\_nas\\_SetNetworkPreference](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SetNetworkPreference\\_t](#) \*reqArg)
- int [unpack\\_nas\\_SetNetworkPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetNetworkPreference\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SetRoamingIndicatorCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetRoamingIndicatorCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SetServingSystemCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetServingSystemCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_SlqsGetLTECphyCAInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SlqsGetLTECphyCAInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSSetSysSelectionPrefCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSetSysSelectionPrefCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasSwiOTAMessageCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasSwiOTAMessageCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSInitiateNetworkRegistration](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSInitiateNetworkRegistration\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSInitiateNetworkRegistration](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SLQSNasSwiOTAMessageCallback](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasSwiOTAMessageCallback\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasSwiOTAMessageCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_nas\\_SLQSGetPLMNName](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSGetPLMNName\\_t](#) \*reqArg)
- int [unpack\\_nas\\_SLQSGetPLMNName](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetPLMNName\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasGetCellLocationInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNasGetCellLocationInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetCellLocationInfo\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetNetworkTime](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSGetNetworkTime](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetNetworkTime\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasNetworkTimeCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasNetworkTimeCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SetNasLTECphyCalndCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetNasLTECphyCalndCallback\\_ind\\_t](#) \*pOutput)

## 9.6.1 Macro Definition Documentation

9.6.1.1 #define NAS\_MAX\_DESCRIPTION\_LENGTH 255

9.6.1.2 #define NAS\_MAX\_NUM\_NETWORKS 30

9.6.1.3 #define NAS\_OTA\_MESSAGE\_MAX\_BUF\_SIZE 2048

9.6.1.4 `#define NAS_PLMN_LENGTH 3`

9.6.1.5 `#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255`

## 9.6.2 Enumeration Type Documentation

9.6.2.1 `enum LIBPACK_NAS_LTE_CPHY_CA_BW_NRB`

Enumerator

*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6*  
*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15*  
*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25*  
*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50*  
*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75*  
*eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100*

9.6.2.2 `enum LIBPACK_NAS_LTE_CPHY_SCELL_STATE`

Enumerator

*eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED*  
*eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED*  
*eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED*

9.6.2.3 `enum NAS_LTE_CPHY_CA_BW_NRB_LITE`

Enumerator

*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6*  
*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15*  
*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25*  
*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50*  
*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75*  
*eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100*

9.6.2.4 `enum NAS_LTE_CPHY_SCELL_STATE_LITE`

Enumerator

*eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED\_LITE*  
*eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED\_LITE*  
*eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED\_LITE*

## 9.6.3 Function Documentation

9.6.3.1 `int pack_nas_GetACCOLC ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Parameters

<i>in, out</i>	<i>pCtx</i>	qmi request context
<i>out</i>	<i>pReq</i>	qmi request buffer
<i>out</i>	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.6.3.2** int pack\_nas\_GetANAAAuthenticationStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.6.3.3** int pack\_nas\_GetCDMANetworkParameters ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.6.3.4** int pack\_nas\_GetHomeNetwork ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

get home network pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

## See Also

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

**9.6.3.5** `int pack_nas_GetNetworkPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**9.6.3.6** `int pack_nas_GetRFInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get rf info pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.7** `int pack_nas_GetServingNetwork ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.8** `int pack_nas_GetServingNetworkCapabilities ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

### 9.6.3.9 int pack\_nas\_GetSignalStrengths ( pack\_qmi\_t \* pCtx, uint8\_t \* pReq, uint16\_t \* pLen )

get signal strengths pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

### 9.6.3.10 int pack\_nas\_PerformNetworkScan ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

### 9.6.3.11 int pack\_nas\_SetACCOLC ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_nas\_SetACCOLC\_t reqParam )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

## Returns

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

## See Also

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

**9.6.3.12** `int pack_nas_SetLURejectCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

## Returns

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

## See Also

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

**9.6.3.13** `int pack_nas_SetNetworkPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SetNetworkPreference_t * reqArg )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pack</i>	default prototype

**9.6.3.14** `int pack_nas_SetRFInfoCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>bEnable</i>	0/1 value to disable/enable indication respectively

## Returns

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

## See Also

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

**9.6.3.15** `int pack_nas_SlqsGetLTECphyCAInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.6.3.16 `int pack_nas_SLQSGetNetworkTime ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

9.6.3.17 `int pack_nas_SLQSGetPLMNName ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSGetPLMNName_t * reqArg )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request prarmeters

## Returns

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

## See Also

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

9.6.3.18 `int pack_nas_SLQSGetServingSystem ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.19** `int pack_nas_SLQSGetSignalStrength ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint16_t reqMask )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqMask</i>	request mask for fetching extra signal info

#### Returns

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

#### See Also

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

**9.6.3.20** `int pack_nas_SLQSGetSysInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

**9.6.3.21** `int pack_nas_SLQSGetSysSelectionPref ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

**9.6.3.22** `int pack_nas_SLQSIInitiateNetworkRegistration ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSIInitiateNetworkRegistration_t * pReqParam )`

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request Parameters

**Returns**

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

**See Also**

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

**9.6.3.23** int pack\_nas\_SLQSNasConfigSigInfo2 ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_nas\_SLQSNasConfigSigInfo2\_t \* *pReqParam* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

**Returns**

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

**See Also**

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

**9.6.3.24** int pack\_nas\_SLQSNasGetCellLocationInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**9.6.3.25** int pack\_nas\_SLQSNasGetSigInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

get sig info pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.26** int pack\_nas\_SLQSNasIndicationRegisterExt ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_SLQSNasIndicationRegisterExt\_t \* *pReqParam* )

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

## Returns

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

## See Also

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

**9.6.3.27** int pack\_nas\_SLQSNasSwtModemStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.28** int pack\_nas\_SLQSNasSwtOTAMessageCallback ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_SLQSNasSwtOTAMessageCallback\_t \* *pReqParam* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request prarmeters

**Returns**

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

**See Also**

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

**9.6.3.29** int pack\_nas\_SLQSSetBandPreference ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, uint64\_t *bandPref* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>band</i>	preference

**Returns**

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

**See Also**

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

**9.6.3.30** int pack\_nas\_SLQSSetSignalStrengthsCallback ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_SLQSSetSignalStrengthsCallback\_t \* *pReqParam* )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request prarmeters

**Returns**

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

**See Also**

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

**9.6.3.31** int pack\_nas\_SLQSSetSysSelectionPref ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_SLQSSetSysSelectionPref\_t \* *pReqParam* )

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request Parameters

## Returns

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

## See Also

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

**9.6.3.32** int pack\_nas\_SLQSSwiGetLteCQI ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

**9.6.3.33** int unpack\_nas\_GetACCOLC ( uint8\_t \* *pResp*, uint16\_t *respLen*, uint8\_t \* *pAccolc* )

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pAccolc</i>	accolc

## Returns

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

## See Also

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

**9.6.3.34** int unpack\_nas\_GetANAAAAAuthenticationStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, uint32\_t \* *pAuthStatus* )

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>authStatus</i>	auth status

**Returns**

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

**See Also**

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

**9.6.3.35** `int unpack_nas_GetCDMANetworkParameters ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_GetCDMANetworkParameters_t * pOutput )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	qmi output parameters

**Returns**

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

**See Also**

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

**9.6.3.36** `int unpack_nas_GetHomeNetwork ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t *  
pOutput )`

get home network unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.6.3.37** `int unpack_nas_GetNetworkPreference ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetNetwork-  
Preference_t * pOutput )`

**9.6.3.38** `int unpack_nas_GetRFInfo ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetRFInfo_t * pOutput )`

get rf info unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

**9.6.3.39** `int unpack_nas_GetServingNetwork ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

**9.6.3.40** `int unpack_nas_GetServingNetworkCapabilities ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetworkCapabilities_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

**9.6.3.41** `int unpack_nas_GetSignalStrengths ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetSignalStrengths_t * pOutput )`

get signal strengths unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.6.3.42** `int unpack_nas_PerformNetworkScan ( uint8_t * pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t * pOutput )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.6.3.43** `int unpack_nas_SetACCOLC ( uint8_t * pResp, uint16_t respLen )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

**9.6.3.44** `int unpack_nas_SetDataCapabilitiesCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetDataCapabilitiesCallback_ind_t * pOutput )`

Data Capabilities indication unpack

**Parameters**

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

**Returns**

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

## See Also

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

9.6.3.45 `int unpack_nas_SetEventReportInd ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t * pOutput )`

9.6.3.46 `int unpack_nas_SetLURjectCallback ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.6.3.47 `int unpack_nas_SetNasLTECphyCalIndCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCalIndCallback_ind_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

## Returns

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

## See Also

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

9.6.3.48 `int unpack_nas_SetNetworkPreference ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t * pOutput )`

9.6.3.49 `int unpack_nas_SetRFInfoCallback ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.6.3.50 `int unpack_nas_SetRoamingIndicatorCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetRoamingIndicatorCallback_ind_t * pOutput )`

Roaming indication unpack

## Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

## Returns

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

## See Also

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

9.6.3.51 `int unpack_nas_SetServingSystemCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t * pOutput )`

9.6.3.52 `int unpack_nas_SlqsGetLTECphyCAInfo ( uint8_t * pResp, uint16_t respLen, unpack_nas_SlqsGetLTECphyCAInfo_t * pOutput )`

9.6.3.53 `int unpack_nas_SLQSGetNetworkTime ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetNetworkTime_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.6.3.54 `int unpack_nas_SLQSGetPLMNName ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetPLMNName_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

**9.6.3.55** `int unpack_nas_SLQSGetservingSystem ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetservingSystem_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

**9.6.3.56** `int unpack_nas_SLQSGetSignalStrength ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

**9.6.3.57** `int unpack_nas_SLQSGetSysInfo ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.6.3.58 `int unpack_nas_SLQSGetSysSelectionPref ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSGetSysSelectionPref_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.6.3.59 `int unpack_nas_SLQSInitiateNetworkRegistration ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.6.3.60 `int unpack_nas_SLQSNasConfigSigInfo2 ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.6.3.61 `int unpack_nas_SLQSNasGetCellLocationInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasGetCellLocationInfo_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.6.3.62 int unpack\_nas\_SLQSNasGetSigInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSNasGetSigInfo\_t \* *pOutput* )

get sig info unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.6.3.63 int unpack\_nas\_SLQSNasIndicationRegisterExt ( uint8\_t \* *pResp*, uint16\_t *respLen* )

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.6.3.64 int unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t \* *pOutput* )

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

**Returns**

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

**See Also**

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

9.6.3.65 `int unpack_nas_SLQSNasSigInfoCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasSigInfoCallback_ind_t * pOutput )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	sig info indication unpacked

**Returns**

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

**See Also**

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

9.6.3.66 `int unpack_nas_SLQSNasSwtModemStatus ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasSwtModemStatus_t * pOutput )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

9.6.3.67 `int unpack_nas_SLQSNasSwtOTAMessageCallback ( uint8_t * pResp, uint16_t respLen )`

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

## See Also

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

9.6.3.68 `int unpack_nas_SLQSNasSwiOTAMessageCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t * pOutput )`

OTA message indication unpack

## Parameters

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

## Returns

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

## See Also

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

9.6.3.69 `int unpack_nas_SLQSNasSysInfoCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSysInfoCallback_ind_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.6.3.70 `int unpack_nas_SLQSSetBandPreference ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

9.6.3.71 `int unpack_nas_SLQSSetSignalStrengthsCallback ( uint8_t * pResp, uint16_t respLen )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

### 9.6.3.72 int unpack\_nas\_SLQSSetSysSelectionPref ( uint8\_t \* *pResp*, uint16\_t *respLen* )

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

### 9.6.3.73 int unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t \* *pOutput* )

System Selection Preference indication unpack

**Parameters**

in	<i>pResp</i>	qmi indication from modem
in	<i>respLen</i>	qmi indication length
out	<i>pOutput</i>	indication unpacked

**Returns**

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

**See Also**

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

### 9.6.3.74 int unpack\_nas\_SLQSSwiGetLteCQI ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSSwiGetLteCQI\_t \* *pOutput* )

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

## See Also

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

## 9.7 qaCbkCatEventReportInd.h File Reference

### Data Structures

- struct [CatEventIDDataTlv](#)
- struct [CatAlphaIdentifierTlv](#)
- 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 [QmiCbkCatEventStatusReportInd](#) \*pAipResp)

#### 9.7.1 Macro Definition Documentation

9.7.1.1 #define [QMI\\_CAN\\_COMMON\\_EVENT\\_TLV\\_NUMBER](#) 11

9.7.1.2 #define [QMI\\_MAX\\_CAT\\_EVENT\\_DATA\\_LENGTH](#) 255

## 9.7.2 Enumeration Type Documentation

### 9.7.2.1 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
```

### 9.7.2.2 enum eQMI\_CAT\_EVENT\_REPORT\_IND\_TLV\_LENGTH

Enumerator

```
eTLV_SETUP_EVENT_LIST_LENGTH
eTLV_REFRESH_LENGTH
eTLV_END_PROACTIVE_SESSION_LENGTH
```

## 9.7.3 Function Documentation

9.7.3.1 enum eQCWWANError UpkQmiCbkCatEventReportInd ( BYTE \* *pMdmResp*, struct QmiCbkCatEventStatusReportInd \* *pAipResp* )

## 9.8 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)

### 9.8.1 Macro Definition Documentation

9.8.1.1 `#define QMI_SWIOMA_DM_CONFIG 0x01`

9.8.1.2 `#define QMI_SWIOMA_DM_FOTA 0x00`

9.8.1.3 `#define QMI_SWIOMA_DM_NOT 0x02`

### 9.8.2 Enumeration Type Documentation

9.8.2.1 enum [eQMI\\_SWIOMA\\_DM\\_EVENT\\_REPORT\\_IND](#)

Enumerator

***eTLV\_IND\_OMA\_DM\_FOTA***  
***eTLV\_IND\_OMA\_DM\_CONFIG***  
***eTLV\_IND\_OMA\_DM\_NOT***

### 9.8.3 Function Documentation

9.8.3.1 enum [eQCWWANError UpkQmiCbkSwiOmaDmEventReportInd](#) ( BYTE \* *pMdmResp*, struct [QmiCbkSwiOmaDmEventStatusReportInd](#) \* *pApiResp* )

9.8.3.2 package enum [eQCWWANError UpkQmiCbkSwiOmaDmEventReportIndExt](#) ( BYTE \* *pMdmResp*, struct [QmiCbkSwiOmaDmEventStatusReportInd](#) \* *pApiResp* )

## 9.9 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](#) \*pGetAudioVolTLBCfgReq, [GetAudioVolTLBConfigResp](#) \*pGetAudioVolTLBCfgResp)
- [ULONG SLQSSetAudioVolTLBConfig](#) ([SetAudioVolTLBConfigReq](#) \*pSetAudioVolTLBCfgReq, [SetAudioVolTLBConfigResp](#) \*pSetAudioVolTLBCfgResp)

### 9.9.1 Detailed Description

Audio Service API function prototypes.

### 9.9.2 Function Documentation

#### 9.9.2.1 [ULONG SLQSGetAudioPathConfig](#) ( [GetAudioPathConfigReq](#) \* *pGetAudioPathConfigReq*, [GetAudioPathConfigResp](#) \* *pGetAudioPathConfigResp* )

This API gets the audio path configuration parameters.

##### Parameters

<i>pGetAudioPathConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">GetAudioPathConfigReq</a> for more information</li> </ul>
<i>pGetAudioPathConfigResp</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

#### 9.9.2.2 [ULONG SLQSGetAudioProfile](#) ( [GetAudioProfileReq](#) \* *pGetAudioProfileReq*, [GetAudioProfileResp](#) \* *pGetAudioProfileResp* )

This API get the profile content of the requested audio generator.

## Parameters

<i>pGetAudioProfileReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioProfileReq</a> for more information</li> </ul>
<i>pGetAudioProfileResp</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

### 9.9.2.3 ULONG SLQSGetAudioVolTLBConfig ( GetAudioVolTLBConfigReq \* pGetAudioVolTLBCfgReq, GetAudioVolTLBConfigResp \* pGetAudioVolTLBCfgResp )

This API gets the audio path configuration parameters.

## Parameters

<i>pGetAudioVolTLBCfgReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioVolTLBConfigReq</a> for more information</li> </ul>
<i>pGetAudioVolTLBCfgResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">GetAudioVolTLBConfigResp</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

### 9.9.2.4 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

### 9.9.2.5 ULONG SLQSSetAudioProfile ( SetAudioProfileReq \* pSetAudioProfileReq )

This API sets an audio profile.

**Parameters**

<i>pSetAudio-ProfileReq[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

### 9.9.2.6 ULONG SLQSSetAudioVoTLBConfig ( SetAudioVoTLBConfigReq \* pSetAudioVoTLBConfigReq, SetAudioVoTLBConfigResp \* pSetAudioVoTLBConfigResp )

This API sets the audio path configuration parameters.

**Parameters**

<i>pSetAudioVoTL-BCfgReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SetAudioVoTLBConfigReq</a> for more information</li> </ul>
<i>pSetAudioVoTL-BCfgResp[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

## 9.10 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)

### 9.10.1 Detailed Description

Card Application Toolkit API function headers.

### 9.10.2 Function Documentation

#### 9.10.2.1 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 pData in bytes</li></ul>
<i>pData[IN]</i>	<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

**9.10.2.2 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[IN]</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

**9.11 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 [voiceSetAllCallStatusCbkInfo](#)
- struct [\\_transLayerInfoNotification](#)
- struct [\\_transNWRRegInfoNotification](#)
- 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 [imsSMSCConfigInfo](#)
- struct [imsUserConfigInfo](#)
- struct [imsVoIPConfigInfo](#)

- struct [USSDNoWaitIndicationInfo](#)
- struct [channelRate](#)
- struct [DUNCallInfoInd](#)
- struct [SMSAsyncRawSend\\_s](#)
- struct [LteNasReleaseInfo\\_s](#)
- struct [SwiOTAMsg\\_s](#)
- struct [DHCPOption](#)
- struct [DHCPOptionList](#)
- struct [WdsDHCPv4ClientLeaseInd](#)
- 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 [t\\_Sv](#)
- struct [t\\_sensor](#)
- struct [t\\_gpsTime](#)
- struct [QmiCbkLocBestAvailPosInd](#)
- struct [imsaRegStatusInfo](#)
- struct [imsaSvcStatusInfo](#)
- struct [imsaRatStatusInfo](#)
- struct [imsaPdpStatusInfo](#)
- struct [satelliteInfo](#)
- struct [gnssSvInfoNotification](#)
- struct [delAssistDataStatus](#)
- struct [QmiCbkNasLTECphyCalInfo](#)
- struct [RankIndicatorInd](#)
- struct [QmiCbkLocInjectUTCTimeInd](#)
- struct [QmiCbkLocInjectPositionInd](#)
- struct [UIMSlotStatusChangeInfo](#)
- struct [QmiCbkLocEngineStateInd](#)
- struct [\\_getResetInfoNotification](#)
- struct [\\_MitigationDevInfo](#)
- struct [QmiCbkTmdMitiLvlRptInd](#)
- struct [QmiCbkLocSetExtPowerConfigInd](#)

## 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_PHY_SLOT_STATUS` 0x00000010
- #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 `DHCP_MAX_NUM_OPTIONS` 30
- #define `DHCP_OPTION_DATA_BUF_SIZE` 2048 /\* current max size of raw message in SDK process is 2048 \*/
- #define `MAX_MITIGATION_DEV_ID_LEN` 255
- #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 radioInterface)
- typedef void(\* `tFNRFRInfo` )(ULONG radioInterface, ULONG activeBandClass, ULONG activeChannel)
- typedef void(\* `tFNLURreject` )(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(\* [tFNAIICallStatus](#) )(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, BYTE session\_status, ULONG pos\_src)
- typedef void(\* tFNNewRMTransferStatistics )(QmiCbkWdsStatisticsIndState \*pMsg)
- typedef void(\* tFNDHCPv4ClientLeaseStatus )(BYTE instance, WdsDHCPv4ClientLeaseInd \*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(\* tFNBestAvailPos )(QmiCbkLocBestAvailPosInd \*pBestAvailPos)
- 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)
- typedef void(\* tFNRankIndicator )(RankIndicatorInd \*pRankIndicatorInd)
- typedef void(\* tFNInjectUTCTime )(QmiCbkLocInjectUTCTimeInd \*pInjectUTCTimeNotification)
- typedef void(\* tFNInjectPosition )(QmiCbkLocInjectPositionInd \*pInjectPositionNotification)
- typedef void(\* tFNCbkUimSlotStatusChangeInd )(UIMSlotStatusChangeInfo \*pQmiCbkUimSlotStatusChangeInd)
- typedef void(\* tFNSetEngineState )(QmiCbkLocEngineStateInd \*pSetLocEngineState)
- typedef struct  
    \_getResetInfoNotification ResetInfoNotification
- typedef void(\* tFNResetInfo )(ResetInfoNotification \*pResetInfoNotification)
- typedef void(\* tFNMTLvlRpt )(QmiCbkTmdMtlLvlRptInd \*pSetLocCradleMount)
- typedef void(\* tFNSetExtPowerConfig )(QmiCbkLocSetExtPowerConfigInd \*pSetExtConfigIndStatus)

## 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 SetFwDIdCompletionCb` (`tFNFwDIdCompletion` 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](#) (tFNAllCallStatus pCallback)
- [ULONG SLQSSetTransLayerInfoCallback](#) (tFNtransLayerInfo pCallback)
- [ULONG SLQSSetTransNWRegInfoCallback](#) (tFNtransNWRegInfo pCallback)
- [ULONG SLQSSetSysSelectionPrefCallBack](#) (tFNSysSelectionPref pCallback)
- [ULONG SLQSUIIMSetRefreshCallBack](#) (tFNUIMRefresh pCallback)
- [ULONG SLQSUIIMSetStatusChangeCallBack](#) (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 SLQSSetSwtHDPersCallback](#) (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](#) (NasSwtIndReq \*req, tFNASwtOTAMsg pCallback)
- [ULONG SetGPSCallback](#) (tFNNewGPS pCallback)
- [ULONG SetRMTransferStatisticsCallback](#) (tFNNewRMTransferStatistics pCallback)
- [ULONG SLQSSetDHCPv4ClientLeaseStatusCallback](#) (BYTE instance, tFNDHCPv4ClientLeaseStatus 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 SLQSSetIMSAStatusCallback](#) (tFNImsaRatStatus pCallback)
- [ULONG SLQSSetIMSApdpStatusCallback](#) (tFNImsaPdpStatus pCallback)
- [ULONG SLQSNasSigInfo2CallBack](#) (tFNSigInfo pCallback, [setSignalStrengthInfo](#) \*pSigInfo2)
- [ULONG SetLocGnssSvInfoCallback](#) (tFNGnssSvInfo pCallback)
- [ULONG SetLocDeleteAssistDataCallback](#) (tFNDeAssistData pCallback)
- [ULONG SetNasLTECphyCaIndCallback](#) (tFNASwiLTECphyCallInfo pCallback)
- [ULONG SetRankIndicatorCallback](#) (tFNRankIndicator pCallback)
- [ULONG SLQSSetLocInjectUTCTimeCallback](#) (tFNInjectUTCTime pCallback)
- [ULONG SLQSSetLocInjectPositionCallback](#) (tFNInjectPosition pCallback)
- [ULONG SetUimSlotStatusChangeCallback](#) (tFNCbkUimSlotStatusChangeInd pCallback)
- [ULONG SetLocEngineStateCallback](#) (tFNSetEngineState pCallback)
- [ULONG SLQSSetSwiGetResetInfoCallback](#) (tFNResetInfo pCallback)
- [ULONG SLQSTmdMitigationLvIRptCallback](#) (TmdMitigationLvIndReq \*req, tFNMitLvIRpt pCallback)
- [ULONG SetLocSetExtPowerConfigCallback](#) (tFNSetExtPowerConfig pCallback)
- [ULONG SetLocBestAvailPosCallback](#) (tFNBstAvailPos pCallback)

### 9.11.1 Detailed Description

Callback Service API function prototypes.

### 9.11.2 Macro Definition Documentation

9.11.2.1 `#define CBK_DISABLE_EVENT 0x00`

9.11.2.2 `#define CBK_ENABLE_EVENT 0x01`

9.11.2.3 `#define CBK_NOCHANGE 0xFF`

9.11.2.4 `#define DEREGISTER_EVENT 0x00`

9.11.2.5 `#define DEREGISTER_SRV 0x00`

9.11.2.6 `#define DHCP_MAX_NUM_OPTIONS 30`

9.11.2.7 `#define DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */`

9.11.2.8 `#define EVENT_MASK_CARD 0x00000001`

9.11.2.9 `#define EVENT_MASK_DEREGISTER_ALL 0x00000000`

9.11.2.10 `#define EVENT_MASK_PHY_SLOT_STATUS 0x00000010`

9.11.2.11 `#define FIRST_INSTANCE 0x00`

9.11.2.12 `#define INVALID_INSTACNE 0x08`

9.11.2.13 `#define IPV4 4`

9.11.2.14 `#define IPV4V6 7`

9.11.2.15 `#define IPV6 6`

9.11.2.16 `#define LOC_EVENT_MASK_ENG_STATE 0x00000080`

9.11.2.17 `#define LOC_EVENT_MASK_GNSS_SV_INFO 0x00000002`

9.11.2.18 `#define LOC_EVENT_MASK_INJECT_TIME 0x00000010`

9.11.2.19 `#define LOC_EVENT_MASK_SENSOR_STREAM 0x00000400`

9.11.2.20 `#define LOC_EVENT_MASK_TIME_SYNC 0x00000800`

9.11.2.21 `#define LOC_EVENT_POSITION_REPORT 0x00000001`

9.11.2.22 `#define MAX_MITIGATION_DEV_ID_LEN 255`

9.11.2.23 `#define MAX_NO_OF_APPLICATIONS 10`

9.11.2.24 `#define MAX_NO_OF_CALLS 20`

9.11.2.25 `#define MAX_NO_OF_FILES 255`

9.11.2.26 `#define MAX_NO_OF_SLOTS 5`

9.11.2.27 `#define MAX_NO_OF_UUSINFO 20`

9.11.2.28 `#define MAX_PATH_LENGTH 255`

9.11.2.29 `#define MAX_RADIO_INTERFACE_LIST 255`

9.11.2.30 `#define MAXUSSDLENGTH 182`

9.11.2.31 `#define NAS_SRV 0x02`

9.11.2.32 `#define NUM_OF_SET 0xFF`

9.11.2.33 `#define PDS_SRV 0x04`

9.11.2.34 `#define QMI_ETWS_MAX_PAYLOAD_LENGTH 1254 /* Qualcomm defined max */`

9.11.2.35 `#define QMI_MAX_VOICE_NUMBER_LENGTH 81`

9.11.2.36 `#define QMI_WMS_MAX_PAYLOAD_LENGTH 256`

9.11.2.37 `#define REGISTER_EVENT 0x01`

9.11.2.38 `#define REGISTER_SRV 0x01`

9.11.2.39 `#define SECOND_INSTANCE 0x01`

9.11.2.40 `#define SIGSTRENGTH_THRESHOLD_ARR_SZ 5`

9.11.2.41 `#define THIRD_INSTANCE 0x02`

9.11.2.42 `#define USSD_DCS_8BIT 0x02 /* 8-bit coding scheme */`

9.11.2.43 `#define USSD_DCS_ASCII 0x01 /* ASCII coding scheme */`

9.11.2.44 `#define USSD_DCS_UCS2 0x03 /* UCS2 coding scheme */`

9.11.2.45 `#define VOICE_SRV 0x08`

9.11.2.46 `#define WDS_SRV 0x01`

### 9.11.3 Typedef Documentation

#### 9.11.3.1 `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>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>

#### 9.11.3.2 `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>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>

## 9.11.3.3 typedef enum device\_state\_enum eDevState

Device State enumeration

- See [device\\_state\\_enum](#) for more details

## 9.11.3.4 typedef enum SMSEventType eSMSEventType

This enumeration defines the different type of SMS events that are received

- See [SMSEventType](#) for more details

## 9.11.3.5 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>

## 9.11.3.6 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>

### 9.11.3.7 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 <ul style="list-style-type: none"> <li>0x01 - Ready to accept sensor data</li> <li>0x00 - Not ready to accept sensor data</li> </ul> </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>

### 9.11.3.8 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>

### 9.11.3.9 typedef struct \_modemTempNotification modemTempNotification

Contains the parameters passed for SLQSSetModemTempCallback by the device.

#### Parameters

<i>ModemTempState</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

## 9.11.3.10 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.

**9.11.3.11 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>

**9.11.3.12 typedef struct \_getResetInfoNotification ResetInfoNotification**

Contains the parameters passed for SLQSSetSwiGetResetInfoCallback by the device.

## Parameters

<i>type</i>	<ul style="list-style-type: none"> <li>• type of reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - warm</li> <li>– 2 - hard</li> <li>– 3 - crash</li> <li>– 4 - power down</li> </ul> </li> </ul>
<i>source</i>	<ul style="list-style-type: none"> <li>• entity which initiated the reset or power down, possible values listed below: <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - user requested</li> <li>– 2 - hardware switch</li> <li>– 3 - temperature critical</li> <li>– 4 - voltage critical</li> <li>– 5 - configuration update</li> <li>– 6 - LWM2M</li> <li>– 7 - OMA-DM</li> <li>– 8 - FOTA</li> </ul> </li> </ul>

## Note

None

9.11.3.13 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

#### 9.11.3.14 typedef union **sessionInfo** sessionInformation

This union **sessionInfo** consist of **omaDmFotaTlv**, **omaDmConfigTlv** and **omaDmNotificationsTlv**, out of which one will be unpacked against pEventFields.

#### 9.11.3.15 typedef union **sessionInfoExt** sessionInformationExt

This union **sessionInfo** consist of **omaDmFotaTlv** and **omaDmConfigTlv**, out of which one will be unpacked against pEventFields.

#### 9.11.3.16 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>

#### 9.11.3.17 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>

#### 9.11.3.18 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>

#### 9.11.3.19 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>

### 9.11.3.20 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 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 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 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 NULL, if this event is not present in the smsEventType parameter Note: None</li> </ul>

### 9.11.3.21 typedef struct **SMSMessageMode** **SMSMessageModeInfo**

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>
--------------------	--

### 9.11.3.22 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 0 - UIM 1 - NV</li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>Index of the new message</li> </ul>

9.11.3.23 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>
-----------------	---

9.11.3.24 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>

9.11.3.25 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>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>

9.11.3.26 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>

9.11.3.27 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>
-------------------------	--

9.11.3.28 `typedef void(* tFNAIICallStatus)(voiceSetAllCallStatusCbKInfo *pVoiceSetAllCallStatusCbKInfo)`

Voice Call Status Callback function. This function pointer will be executed to process received Indication.

## Parameters

<i>pVoiceSetAll- CallStatusCbK- 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="#">voiceSetAllCallStatusCbKInfo</a> for more information.</li> </ul>
--	--

9.11.3.29 `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>
---	--

9.11.3.30 `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>
------------------------------	--

9.11.3.31 `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>
---------------------------	---

9.11.3.32 `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> <li>• 56 - BBC16 - Band Class 16</li> <li>• 57 - BC17 - Band Class 17</li> <li>• 58 - BC18 - Band Class 18</li> <li>• 59 - BC19 - Band Class 19</li> <li>• 60 to 64 - Reserved</li> </ul>
------------------	---

**Note**

Timeout: NA To set the band preference the API [SLQSSetBandPreference\(\)](#) should be used

9.11.3.33 `typedef void(* tFNBstAvailPos)(QmiCbkLocBestAvailPosInd *pBestAvailPos)`

9.11.3.34 `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

9.11.3.35 `typedef void(* tFNCbkUimSlotStatusChangeInd)(UIMSlotStatusChangeInfo *pQmiCbkUimSlotStatusChangeInd)`

Slot Status Change Notification callback.

#### Parameters

<i>pQmiCbkUimSlotStatusChangeInd</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UIMSlotStatusChangeInfo</a> for more information.</li> </ul>
--------------------------------------	---

9.11.3.36 `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>
------------------	--

9.11.3.37 `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>
--------------------------	---

9.11.3.38 `typedef void(* tFNDelAssistData)(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>
---------------------------------	---

9.11.3.39 `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

9.11.3.40 `typedef void(* tFNDHCPv4ClientLeaseStatus)(BYTE instance, WdsDHCPv4ClientLeaseInd *pMsg)`

DHCPv4 client lease status message callback function.

#### Parameters

<i>pMsg[OUT]</i>	<ul style="list-style-type: none"> <li>Events related to DHCPv4 client lease, see <a href="#">WdsDHCPv4ClientLeaseInd</a> for details</li> </ul>
------------------	--

9.11.3.41 `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>
-----------------------------	--

9.11.3.42 `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>
---------------------	--

9.11.3.43 `typedef void(* tFNEventPosition)(QmiCbkLocPositionReportInd *pLocPositionReport)`

9.11.3.44 `typedef void(* tFNFwDidCompletion)(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, the possible return values are listed below:             <ul style="list-style-type: none"> <li>eQCWWAN_ERR_NONE - indicates firmware download/switching is successful</li> <li>eQCWWAN_ERR_SWIIM_FIRMWARE_NOT_DOWNLOADED - indicates no actual download takes place, this is the case of image switching stored on device</li> <li>eQCWWAN_ERR_SWIIM_FW_ENTER_DOWNLOAD_MODE - indicates modem enters firmware download mode, firmware flashing is going to be started.</li> <li>eQCWWAN_ERR_SWIIM_FW_FLASH_COMPLETE - indicates firmware flashing was complete, SDK is waiting for modem to reboot (can be more than one time), when modem is ready, SDK will send eQCWWAN_ERR_NONE to the host application.</li> </ul> </li> </ul>
-------------------	---

**Note**

Does not require communication with the device

**9.11.3.45** `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>
---------------------------------	--

**9.11.3.46** `typedef void(* tFNHDRPersonality)(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

**9.11.3.47** `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>
----------------------------	---

**9.11.3.48** `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>
----------------------------	---

**9.11.3.49** `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>
----------------------------	---

9.11.3.50 `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>
----------------------------	---

9.11.3.51 `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>
------------------------------	---

9.11.3.52 `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>
---------------------------	--

9.11.3.53 `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>
---------------------------	--

9.11.3.54 `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>
----------------------------	---

9.11.3.55 `typedef void(* tFNImSVoIPConfig)(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>
----------------------------	---

9.11.3.56 `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

9.11.3.57 `typedef void(* tFNInjectPosition)(QmiCbKLocInjectPositionInd *pInjectPositionNotification)`

Inject Position Notification callback.

## Parameters

<i>pInjectPosition-Notification</i>	<ul style="list-style-type: none"> <li>See <a href="#">QmiCbKLocInjectPositionInd</a> for more information.</li> </ul>
-------------------------------------	--

9.11.3.58 `typedef void(* tFNInjectSensorData)(QmiCbKLocInjectSensorDataInd *pLocInjectSensorData)`

9.11.3.59 `typedef void(* tFNInjectTimeStatus)(QmiCbKLocInjectTimeInd *pLocInjectTime)`

9.11.3.60 `typedef void(* tFNInjectUTCTime)(QmiCbKLocInjectUTCTimeInd *pInjectUTCTimeNotification)`

Inject UTC Time Notification callback.

## Parameters

<i>pInjectUTCTime-Notification</i>	<ul style="list-style-type: none"> <li>See <a href="#">QmiCbKLocInjectUTCTimeInd</a> for more information.</li> </ul>
------------------------------------	---

### 9.11.3.61 `typedef void(* tFNLUReject)(ULONG serviceDomain, ULONG rejectCause)`

LU reject callback function.

#### Parameters

<i>serviceDomain</i>	<ul style="list-style-type: none"> <li>• Service domain <ul style="list-style-type: none"> <li>– 1 - Circuit Switched</li> <li>– 2 - Packet Switched</li> <li>– 3 - Circuit and Packet Switched</li> </ul> </li> </ul>
<i>rejectCause</i>	<ul style="list-style-type: none"> <li>• Reject cause</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 2 - IMSI unknown in HLR</li> <li>– 3 - Illegal MS</li> <li>– 4 - IMSI unknown in VLR</li> <li>– 5 - IMEI not accepted</li> <li>– 6 - Illegal ME</li> <li>– 11 - PLMN not allowed\</li> <li>– 12 - Location Area not allowed</li> <li>– 13 - Roaming not allowed in this location area</li> <li>– 15 - No Suitable Cells In Location Area</li> <li>– 17 - Network failure</li> <li>– 20 - MAC failure</li> <li>– 21 - Synch failure</li> <li>– 22 - Congestion</li> <li>– 23 - GSM authentication unacceptable</li> <li>– 25 - Not authorized for this CSG</li> <li>– 32 - Service option not supported</li> <li>– 33 - Requested service option not subscribed</li> <li>– 34 - Service option temporarily out of order</li> <li>– 38 - Call cannot be identified</li> <li>– 48 to 63 - retry upon entry into a new cell</li> <li>– 95 - Semantically incorrect message</li> <li>– 96 - Invalid mandatory information</li> <li>– 97 - Message type non-existent or not implemented</li> <li>– 98 - Message type not compatible with the protocol state</li> <li>– 99 - Information element non-existent or not implemented</li> <li>– 100 - Conditional IE error</li> <li>– 101 - Message not compatible with the protocol state</li> <li>– 111 - Protocol error, unspecified</li> </ul> </li> <li>– Note - Any other value received by the mobile station shall be treated as 34, 'Service option temporarily out of order'.</li> <li>* Any other value received by the network shall be treated as 111, 'Protocol error, unspecified'.</li> </ul> <p>See 3GPP TS 24.008, Section 4.4.4.7 and Section 10.5.3.6 See <a href="#">qaGobiApiTableCall-EndReasons.h</a> for Call End reasons</p>

## Note

Technology Supported: UMTS

9.11.3.62 `typedef void(* tFNMemoryFull)(SMSMemoryInfo *pSMSMemoryFullInfo)`

SMS Memory related callback function.

## Parameters

<i>pSMSMemoryFullInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• pointer to <a href="#">SMSMemoryInfo</a>.</li> <li>• see <a href="#">SMSMemoryInfo</a> for details.</li> </ul>
---------------------------------	---

9.11.3.63 `typedef void(* tFNMessageWaiting)(msgWaitingInfo *pSMSMessageWaitingInfo)`

SMS Memory related callback function.

## Parameters

<i>pSMSMessageWaitingInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• pointer to <a href="#">msgWaitingInfo</a>.</li> <li>• see <a href="#">msgWaitingInfo</a> for details.</li> </ul>
-------------------------------------	---

9.11.3.64 `typedef void(* tFNMitlVlRpt)(QmiCbKtmdMitlVlRptInd *pSetLocCradleMount)`

9.11.3.65 `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>
------------------	---

9.11.3.66 `typedef void(* tFNModemTempInfo)(modemTempNotification *pModemTempNotification)`

Modem Temperature Information callback.

## Parameters

<i>pModemTempNotification</i>	<ul style="list-style-type: none"> <li>• See <a href="#">modemTempNotification</a> for more information.</li> </ul>
-------------------------------	---

9.11.3.67 `typedef void(* tFNNet)(ULONG q_depth, BYTE isThrottle, BYTE instancelid)`

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>instancelid</i>	<ul style="list-style-type: none"> <li>qmi instance id</li> </ul>

Note

Does not require communication with the device

9.11.3.68 `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>
------------------------	--

9.11.3.69 `typedef void(* tFNNewGPS)(double dLongitude, double dLatitude, BYTE session_status, ULONG pos_src)`

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>

<i>session_status</i> [I-N]	<ul style="list-style-type: none"> <li>• Session Status <ul style="list-style-type: none"> <li>– 0 - Success</li> <li>– 1 - In progress</li> <li>– 2 - General failure</li> <li>– 3 - Timeout</li> <li>– 4 - User ended the session</li> <li>– 5 - Bad parameter</li> <li>– 6 - Phone is offline</li> <li>– 7 - Engine is locked</li> <li>– 8 - E911 session in progress</li> </ul> </li> </ul>
<i>pos_src</i> [I-N]	<ul style="list-style-type: none"> <li>• position source</li> <li>• Bitmasks <ul style="list-style-type: none"> <li>– 0x01 - GPS</li> <li>– 0x02 - Cell ID</li> <li>– 0x04 - GLONASS</li> <li>– 0x08 - Network</li> <li>– 0x10 - External position injection</li> <li>– Others - unknown</li> </ul> </li> </ul>

#### 9.11.3.70 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>
--------------	---

#### 9.11.3.71 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>
------------------	--

#### 9.11.3.72 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 0 - UIM 1 - NV</li> </ul>
<i>messageIndex</i>	<ul style="list-style-type: none"> <li>Index of the new message</li> </ul>

#### 9.11.3.73 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

9.11.3.74 `typedef void(* tFNOpMode)(ULONG mode)`

9.11.3.75 `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

9.11.3.76 `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>
--------------------------	--

9.11.3.77 `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>

9.11.3.78 `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

9.11.3.79 `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

9.11.3.80 `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>
---------------	---

9.11.3.81 `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>
--------------	--

9.11.3.82 `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>

9.11.3.83 `typedef void( * tFNRankIndicator)(RankIndicatorInd *pRankIndicatorInd)`

9.11.3.84 `typedef void(* tFNResetInfo)(ResetInfoNotification *pResetInfoNotification)`

Get Reset Info Indication callback.

## Parameters

<i>pResetInfo-Notification</i>	<ul style="list-style-type: none"> <li>• See <a href="#">ResetInfoNotification</a> for more information.</li> </ul>
--------------------------------	---

#### 9.11.3.85 `typedef void(* tFNRFInfo)(ULONG radiolInterface, ULONG activeBandClass, ULONG activeChannel)`

RF information callback function.

## Parameters

<i>radiolInterface</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>

#### 9.11.3.86 `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>
----------------	---

#### 9.11.3.87 `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

9.11.3.88 `typedef void(* tFNSensorStreaming)(QmiCbKLocSensorStreamingInd *pLocSensorStream)`

9.11.3.89 `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>
-----------------------	---

9.11.3.90 `typedef void(* tFNSetCradleMount)(QmiCbKLocCradleMountInd *pSetLocCradleMount)`

9.11.3.91 `typedef void(* tFNSetEngineState)(QmiCbKLocEngineStateInd *pSetLocEngineState)`

9.11.3.92 `typedef void(* tFNSetEventTimeSync)(QmiCbKLocEventTimeSyncInd *pSetLocEventTimeSync)`

9.11.3.93 `typedef void(* tFNSetExtPowerConfig)(QmiCbKLocSetExtPowerConfigInd *pSetExtConfigIndStatus)`

9.11.3.94 `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>
--------------------	--

9.11.3.95 `typedef void(* tFNSignalStrength)(INT8 signalStrength, ULONG radiolInterface)`

Signal strength callback function.

#### Parameters

<i>signalStrength</i>	<ul style="list-style-type: none"> <li>Received signal strength (in dBm)</li> </ul>
<i>radiolInterface</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>

9.11.3.96 `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>

9.11.3.97 `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>

9.11.3.98 `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>
---------------------------	---

9.11.3.99 `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>
----------------------------------	--

9.11.3.100 `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>
----------------------	---

**9.11.3.101** `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>
-------------------------------	---

**9.11.3.102** `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

**9.11.3.103** `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>
--------------------------------	---

**9.11.3.104** `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>
--------------------	--

**9.11.3.105** `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>
----------------------------	--

9.11.3.106 `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>
--------------------------------	--

9.11.3.107 `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>
-------------------------------------	--

9.11.3.108 `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>
--------------------------	---

9.11.3.109 `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>
------------------------------	---

9.11.3.110 `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

9.11.3.111 `typedef void(* tFNUSSDNoWaitIndication)(USSDNoWaitIndicationInfo *pNetworkInfo)`

9.11.3.112 `typedef void(* tFNUSSDRelease)(void)`

USSD releaserecallback function prototype

**Note**

Technology Supported: UMTS

9.11.3.113 `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

9.11.3.114 `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

## 9.11.4 Enumeration Type Documentation

## 9.11.4.1 enum device\_state\_enum

Device State enumeration

- See [device\\_state\\_enum](#) for more details

## Enumerator

***DEVICE\_STATE\_DISCONNECTED***  
***DEVICE\_STATE\_READY***  
***DEVICE\_STATE\_BOOT***

## 9.11.4.2 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***

## 9.11.4.3 enum SMSEventType

This enumeration defines the different type of SMS events that are received

- See [SMSEventType](#) for more details

## 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***

### 9.11.5 Function Documentation

9.11.5.1 **ULONG** iSetCATEventCallback ( **tFNCATEvent** *pCallback* )

9.11.5.2 **ULONG** iSetSignalStrengthCallback ( **tFNSignalStrength** *pCallback* )

9.11.5.3 **ULONG** iSLQSSetDUNCallInfoCallback ( **tFNDUNCallInfo** *pCallback* )

9.11.5.4 **ULONG** iSLQSSetSignalStrengthsCallback ( **tFNSLQSSignalStrengths** *pCallback* )

9.11.5.5 **ULONG** iSLQSSetWdsFirstInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.11.5.6 **ULONG** iSLQSSetWdsSecondInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.11.5.7 **ULONG** iSLQSSetWdsThirdInstEventCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.11.5.8 **ULONG** iSLQSSetWdsXferStatsFirstInstCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.11.5.9 **ULONG** iSLQSSetWdsXferStatsSecondInstCallback ( **tFNSLQSWDSEvent** *pCallback* )

9.11.5.10 **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[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

9.11.5.11 **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

#### 9.11.5.12 ULONG SetDataCapabilitiesCallback ( tFNDDataCapabilities 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</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

Does not require communication with the device

**9.11.5.13 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</i> [IN]	<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

**9.11.5.14 ULONG SetFwDldCompletionCbK ( tFNFwDldCompletion *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</i> [IN]	<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

**9.11.5.15    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

**9.11.5.16    ULONG SetLocBestAvailPosCallback ( tFNBestAvailPos pCallback )**

Enables/disables Best Available Location 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

**9.11.5.17    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

**9.11.5.18   ULONG SetLocDeleteAssistDataCallback ( tFNDeIAssistData pCallback )**

Enables/disables Delete Assist Data callback function. This API is used to receive the SUCCESS/FAILURE status of API [SLQSLOCDeIAssData\(\)](#).

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
----------------------	---

**9.11.5.19   ULONG SetLocEngineStateCallback ( tFNSetEngineState pCallback )**

Sends the GPS State Information event to the control point.

**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

**9.11.5.20   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[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

**9.11.5.21   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[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

**9.11.5.22 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[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

**9.11.5.23 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[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

**9.11.5.24 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[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

**9.11.5.25 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[IN]</i>	<ul style="list-style-type: none"> <li>• Callback function pointer (0-Disable)</li> </ul>
----------------------	---

**9.11.5.26 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[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

**9.11.5.27 ULONG SetLocSetExtPowerConfigCallback ( tFNSetExtPowerConfig pCallback )**

Enables/disables the Set External Power Config 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[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

**9.11.5.28 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[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

**9.11.5.29 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[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**

Technology Supported: CDMA  
Timeout: 2 seconds

**9.11.5.30 ULONG SetNasLTECphyCalndCallback ( tFNASwiLTECphyCallInfo pCallback )**

Enables/disables the LTE NAS CA Info callback function.

**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

#### 9.11.5.31 **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

#### 9.11.5.32 **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

**9.11.5.33    `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

**9.11.5.34    `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

**9.11.5.35    `ULONG SetPDSSStateCallback ( 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

**9.11.5.36 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[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

**9.11.5.37 ULONG SetRankIndicatorCallback ( tFNRankIndicator pCallback )****9.11.5.38 ULONG SetRFInfoCallback ( tFNRFInfo 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[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

#### 9.11.5.39 **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

#### 9.11.5.40 **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\(\)](#) instead

##### 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

#### 9.11.5.41 **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.

#### 9.11.5.42 **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

#### 9.11.5.43 **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

**9.11.5.44    `ULONG SetUimSlotStatusChangeCallback ( tFNCbkUimSlotStatusChangeInd pCallback )`**

Enables/disables Slot Status Change 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

**9.11.5.45    `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</i> [IN]	<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

**9.11.5.46    `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

**9.11.5.47    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

**9.11.5.48    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[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  
This callback is sent when the 3GPP or 3GPP2 network sends time information to the User Equipment.

#### 9.11.5.49 **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

#### 9.11.5.50 **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

**9.11.5.51 ULONG SLQSNasSwtOTAMessageCallback ( NasSwIndReg \* req, tFNASwtOTAMsg pCallback )**

Enables/disables the SLQSNasSwtOTAMessageCallback 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="#">NasSwIndReg</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

**9.11.5.52 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.

#### 9.11.5.53 **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[IN]</i>	<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

#### 9.11.5.54 **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[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: 5 seconds

#### 9.11.5.55 **ULONG SLQSSetDHCPv4ClientLeaseStatusCallback ( BYTE *instance*, tFNDHCPv4ClientLeaseStatus *pCallback* )**

Enables/disables the DHCP Client V4 Lease 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>instance</i> [IN]	<ul style="list-style-type: none"> <li>• QMI instance</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

#### 9.11.5.56 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

#### 9.11.5.57 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

**9.11.5.58    ULONG SLQSSetIMSARatStatusCallback ( tFNImsaRatStatus *pCallback* )**

SLQSSetIMSARatStatusCallback

**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

**9.11.5.59    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

**9.11.5.60    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

**9.11.5.61 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

**9.11.5.62 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

**9.11.5.63 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[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

**9.11.5.64 ULONG SLQSSetLocInjectPositionCallback ( tFNInjectPosition pCallback )**

Enables/disables Inject Position callback function. This API is used to receive the SUCCESS/FAILURE status of API [SLQSLOCInjectPosition\(\)](#).

**Parameters**

<i>pCallback[IN]</i>	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
----------------------	---

**9.11.5.65 ULONG SLQSSetLocInjectUTCTimeCallback ( tFNInjectUTCTime pCallback )**

Enables/disables Inject UTC Time callback function. This API is used to receive the SUCCESS/FAILURE status of API [SLQSLOCInjectUTCTime\(\)](#).

## Parameters

<i>pCallback</i> [IN]	<ul style="list-style-type: none"><li>• Callback function pointer (0-Disable)</li></ul>
-----------------------	---

9.11.5.66 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

9.11.5.67 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

**9.11.5.68    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)

**9.11.5.69    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[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

**9.11.5.70    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

**9.11.5.71    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>
in	<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

**9.11.5.72    ULONG SLQSSetRegMgrConfigCallback ( tFNImRegMgrConfig *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

**9.11.5.73 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

**9.11.5.74 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

#### 9.11.5.75 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</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

#### 9.11.5.76 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.

**9.11.5.77 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[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

**9.11.5.78 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[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

**9.11.5.79 ULONG SLQSSetSwtGetResetInfoCallback ( tFNResetInfo pCallback )**

Reset Info callback.

**Parameters**

<i>pCallback</i>	<ul style="list-style-type: none"> <li>• See <a href="#">tFNResetInfo</a> for more information.</li> </ul>
------------------	--

**9.11.5.80 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

**9.11.5.81 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

**9.11.5.82 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

**9.11.5.83 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[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

**9.11.5.84 ULONG SLQSSetWdsEventCallback ( tFNLSQSWDSEvent 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[IN]</i>	<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.

#### 9.11.5.85 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[IN]</i>	<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.

#### 9.11.5.86 **ULONG** SLQSTmdMitigationLvIRptCallback ( **TmdMitigationLvIRptReq** \* *req*, **tFNMitLvlRpt** *pCallback* )

Thermal Mitigation callback.

**Parameters**

<i>req</i>	<ul style="list-style-type: none"> <li>See <a href="#">TmdMitigationLvIRptReq</a> for more information.</li> </ul>
<i>pCallback</i>	<ul style="list-style-type: none"> <li>See <a href="#">tFNMitLvlRpt</a> for more information.</li> </ul>

#### 9.11.5.87 **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[IN]</i>	<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.

#### 9.11.5.88 **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[IN]</i>	<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

**9.11.5.89 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

**9.11.5.90 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

**9.11.5.91    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.

**9.11.5.92    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.

**9.11.5.93    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.

**9.11.5.94    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.

**9.11.5.95    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

**9.11.5.96 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[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

**9.11.5.97 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[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

## 9.11.5.98 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

## 9.12 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 deviceKeySize, 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 \*sn)
- [ULONG SLQSKillSDKProcess](#) ()
- [ULONG SLQSGetDeviceMode](#) (BYTE \*pDeviceMode)
- [ULONG SLQSStartSrv](#) (BYTE action, BYTE mask)
- [ULONG SLQSSetLoggingMask](#) (BYTE mask)
- [ULONG SLQSGetNetStatistic](#) (struct [NetStats](#) \*pNetStatistic, BYTE instance)

### 9.12.1 Detailed Description

Device Connectivity Service API function prototypes.

### 9.12.2 Macro Definition Documentation

9.12.2.1 `#define LEN 10`

9.12.2.2 `#define PORTNAM_LEN 32`

This structure contains the SLQSGetUsbPortNames Information

#### Parameters

<i>AtCmdPort</i>	[OUT] • Name of AT command port
<i>NmeaPort</i>	[OUT] • Name of NMEA port
<i>DmPort</i>	[OUT] • Name of DM port

#### Note

Technology Supported: UMTS/CDMA  
 Device Supported: MC83x5, MC7700/10/50  
 Timeout: 2 seconds  
 Port names are limited to 32 characters.

### 9.12.3 Function Documentation

9.12.3.1 `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[IN]</i>	• Device path pertaining to the device for which the API is being invoked e.g. /dev/qcqm0.
<i>pDeviceKey[IN]</i>	• Device key pertaining to the device for which the API is being invoked

#### Returns

eQCWWAN\_ERR\_NONE on success, eQCWWAN\_ERR\_NO\_DEVICE otherwise

#### Note

Timeout: 2 seconds

### 9.12.3.2 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 <i>deviceId</i>[256] - Device path (e.g. /dev/qcqmio)            CHAR <i>deviceKey</i>[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

### 9.12.3.3 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

#### 9.12.3.4 **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

#### 9.12.3.5 **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

#### 9.12.3.6 **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.

### 9.12.3.7 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

### 9.12.3.8 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

### 9.12.3.9 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

#### 9.12.3.10 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

#### 9.12.3.11 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.

**9.12.3.12 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

**9.12.3.13 ULONG SLQSStart ( BYTE *modem\_index*, CHAR \* *sn* )**

Create the SDK process and IPC sockets for the Application and SDK processes to communicate over.

**Parameters**

<i>in</i>	<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>
<i>in</i>	<i>sn</i>	This field is optional, it can be serial number or usb path for multiple modem feature, it can be retrieved from sytem command "dmesg" when specified, the modem_index will be mapping to sn or usb path 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

**9.12.3.14 ULONG SLQSSStart\_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

**9.12.3.15 ULONG SLQSSStartSrv ( BYTE action, BYTE mask )**

Registers/deregisters for service with unsolicited notifications

**Parameters**

<i>action, 1</i>	for register, 0 for deregister
<i>mask</i>	<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".

## 9.13 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](#)
- struct [getDyingGaspCfg](#)
- struct [setDyingGaspCfg](#)
- struct [getDyingGaspStatistics](#)
- struct [dmsIndicationRegisterReq](#)
- struct [dmsSwiGetResetInfo](#)
- struct [BandCapabilityResp](#)

**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
- #define [MAX\\_DYING\\_GASP\\_CFG\\_SMS\\_CONTENT\\_LENGTH](#) 160
- #define [MAX\\_DYING\\_GASP\\_CFG\\_SMS\\_NUMBER\\_LENGTH](#) 20

## 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)
- [ULONG SLQSSwiGetDyingGaspCfg](#) ([getDyingGaspCfg](#) \*pConfig)
- [ULONG SLQSSwiSetDyingGaspCfg](#) ([setDyingGaspCfg](#) \*pConfig)
- [ULONG SLQSSwiGetDyingGaspStatistics](#) ([getDyingGaspStatistics](#) \*pStatistics)
- [ULONG SLQSSwiClearDyingGaspStatistics](#) ()
- [ULONG SLQSDmsSwiIndicationRegister](#) ([dmsIndicationRegisterReq](#) \*pIndicationRegisterReq)
- [ULONG SLQSDmsSwiGetResetInfo](#) ([dmsSwiGetResetInfo](#) \*pGetResetInfoResp)
- [ULONG SLQSGetBandCapabilities](#) ([BandCapabilityResp](#) \*pBandCapability)

### 9.13.1 Detailed Description

Device Management Service API function prototypes.

### 9.13.2 Macro Definition Documentation

9.13.2.1 `#define IMGDETAILS_LEN 16`

9.13.2.2 `#define MAX_BUILD_ID_LEN 255`

9.13.2.3 `#define MAX_CUST_ID_LEN 64`

9.13.2.4 `#define MAX_CUST_VALUE_LEN 8`

9.13.2.5 `#define MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160`

9.13.2.6 `#define MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20`

9.13.2.7 `#define MAX_FSN_LENGTH 255`

9.13.2.8 `#define UNIQUE_ID_LEN 16`

### 9.13.3 Typedef Documentation

#### 9.13.3.1 typedef struct custFeaturesInfo custFeaturesInfo

This structure contains current settings of custom features

##### Parameters

<i>GpsEnable[OUT]</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[OUT]</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>pIPFam-Support[OUT]</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[OUT]</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[OUT]</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[OUT]</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[OUT]</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>pDHCPRelayEnabled[OUT]</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[OUT]</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>

#### 9.13.3.2 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>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>

## 9.13.3.3 typedef struct dmsCurrentPRLInfo dmsCurrentPRLInfo

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>

## 9.13.3.4 typedef struct ERIFileparams ERIFileparams

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>

## 9.13.3.5 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</i> [OUT]	<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</i> [OUT]	<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[O-UT]</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>

### 9.13.3.6 typedef struct \_SLQSSwiGetHostDevInfoParams SLQSSwiGetHostDevInfoParams

This structure is used to Get Host Device Information

#### Parameters

<i>bManSize[IN/O-UT]</i>	<ul style="list-style-type: none"> <li>• Host Device Manufacturer String Size</li> </ul>
<i>pManString[OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Manufacturer Name(Optional parameter)</li> <li>• Null terminated ASCII String</li> </ul>
<i>bModelSize[IN/-OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Model String Size</li> </ul>
<i>pModelString[O-UT]</i>	<ul style="list-style-type: none"> <li>• Host Device Model String(Optional parameter)</li> <li>• Null terminated ASCII string.</li> </ul>
<i>bSWVerSize[IN/-OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Software Version String Size</li> </ul>
<i>pSWVerString[-OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Software Version String(Optional parameter)</li> <li>• Null terminated ASCII string</li> </ul>
<i>bPlasmaIDSize[-IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Plasma ID String Size</li> </ul>
<i>pPlasmaID-String[OUT]</i>	<ul style="list-style-type: none"> <li>• Host Device Plasma ID String(Optional parameter)</li> <li>• Null terminated alphanumeric ASCII String.</li> </ul>

### 9.13.3.7 typedef struct \_SLQSSwiGetOSInfoParams SLQSSwiGetOSInfoParams

This structure is used to Get OS Information

## Parameters

<i>bNameSize</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>Size of Operating System Name</li> </ul>
<i>pNameString</i> [OUT]	<ul style="list-style-type: none"> <li>Operating System Name(Optional parameter)</li> <li>Null terminated ASCII string</li> </ul>
<i>bVersionSize</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Operating System Version Size</li> </ul>
<i>pVersionString</i> [OUT]	<ul style="list-style-type: none"> <li>Operating System Version String(Optional parameter)</li> <li>Null terminated ASCII string.</li> </ul>

## 9.13.3.8 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>Optional parameter</li> <li>Pointer to receive String containing the Mobile Equipment Identifier(MEID) of the device.</li> </ul>

## 9.13.3.9 typedef struct \_SLQSSwiSetHostDevInfoParams SLQSSwiSetHostDevInfoParams

This structure is used to Set Host Device Information

## Parameters

<i>bManSize</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Manufacturer String Size</li> </ul>
<i>pManString</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Manufacturer Name(Optional parameter)</li> <li>Null terminated ASCII String</li> </ul>
<i>bModelSize</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Model String Size</li> </ul>
<i>pModelString</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Model String(Optional parameter)</li> <li>Null terminated ASCII string.</li> </ul>
<i>bSWVerSize</i> [IN]	<ul style="list-style-type: none"> <li>Host Device Software Version String Size</li> </ul>

<i>pSWVerString</i> [I-N]	<ul style="list-style-type: none"> <li>• Host Device Software Version String(Optional parameter)</li> <li>• Null terminated ASCII string</li> </ul>
<i>bPlasmaIDSize</i> [I-N]	<ul style="list-style-type: none"> <li>• Host Device Plasma ID String Size</li> </ul>
<i>pPlasmaIDString</i> [I-N]	<ul style="list-style-type: none"> <li>• Host Device Plasma ID String(Optional parameter)</li> <li>• Null terminated alphanumeric ASCII String.</li> </ul>

#### 9.13.3.10 typedef struct \_SLQSSwiSetOSInfoParams SLQSSwiSetOSInfoParams

This structure is used to Set OS Information

##### Parameters

<i>bNameSize</i> [I-N]	<ul style="list-style-type: none"> <li>• Size of Operating System Name</li> </ul>
<i>pNameString</i> [I-N]	<ul style="list-style-type: none"> <li>• Operating System Name(Optional parameter)</li> <li>• Null terminated ASCII string</li> </ul>
<i>bVersionSize</i> [I-N]	<ul style="list-style-type: none"> <li>• Operating System Version Size</li> </ul>
<i>pVersionString</i> [I-N]	<ul style="list-style-type: none"> <li>• Operating System Version String(Optional parameter)</li> <li>• Null terminated ASCII string.</li> </ul>

### 9.13.4 Function Documentation

#### 9.13.4.1 ULONG ActivateAutomatic ( CHAR \* *pActivationCode* )

Requests the device to perform automatic service activation

##### Parameters

<i>pActivationCode</i> [I-N]	<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

#### 9.13.4.2 ULONG GetActivationState ( ULONG \* *pActivationState* )

Returns the device activation state.

## Parameters

<i>pActivationState</i> [OUT]	<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

#### 9.13.4.3 ULONG GetDeviceCapabilities ( ULONG \* *pMaxTXChannelRate*, ULONG \* *pMaxRXChannelRate*, ULONG \* *pDataServiceCapability*, ULONG \* *pSimCapability*, ULONG \* *pRadiolfacesSize*, BYTE \* *pRadiolfaces* )

Gets the device capabilities

## Parameters

<i>pMaxTXChannelRate</i> [OUT]	<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>pMaxRXChannelRate</i> [OUT]	<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>pDataServiceCapability</i> [OUT]	<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</i> [-OUT]	<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>pRadioIfaceSize</i> [IN/OUT]	<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>pRadioIface</i> [OUT]	<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

**9.13.4.4 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</i> [OUT]	<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

#### 9.13.4.5 **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> [OUT]	<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> [OUT]	<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

#### 9.13.4.6 **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

#### 9.13.4.7 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. Please use [SLQSUIReadTransparent\(\)](#)(EF ID: 3F00 7F20 6F07 for 2G card and 3F00 7FFF 6F07 for 3G card) instead for new firmware versions and new modules.

## 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

#### 9.13.4.8 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

#### 9.13.4.9 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

#### 9.13.4.10 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.

#### 9.13.4.11 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>Optional parameter</li> <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>Optional parameter</li> <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

**9.13.4.12 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

**9.13.4.13 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

#### 9.13.4.14 **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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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

#### 9.13.4.15 **ULONG** GetVoiceNumber ( **BYTE** *voiceNumberSize*, **CHAR** \* *pVoiceNumber*, **BYTE** *minSize*, **CHAR** \* *pMIN* )

Returns the voice number in use by the device

##### Parameters

<i>voiceNumberSize</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</i> [OUT]	<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[OUT]</i>	<ul style="list-style-type: none"> <li>Optional Parameter</li> <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

**9.13.4.16 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

**9.13.4.17 ULONG SetPower ( ULONG powerMode )**

Sets the operating mode of the device.

**Parameters**

<i>powerMode[IN]</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**

Timeout: 2 seconds

**9.13.4.18 ULONG SLQSDmsSwiGetResetInfo ( dmsSwiGetResetInfo \* pGetResetInfoResp )**

This function is used to get reset info

**Parameters**

<i>pGetResetInfo-Resp</i>	<ul style="list-style-type: none"> <li>See <a href="#">dmsSwiGetResetInfo</a> for more information of the input structure</li> </ul>
---------------------------	--

**9.13.4.19 ULONG SLQSDmsSwiIndicationRegister ( dmsIndicationRegisterReq \* pIndicationRegisterReq )**

This function used to set Swi Indication Register

**Parameters**

<i>pConfig</i>	<ul style="list-style-type: none"> <li>See <a href="#">dmsIndicationRegisterReq</a> for more information of the input structure</li> </ul>
----------------	--

**9.13.4.20 ULONG SLQSGetBandCapabilities ( BandCapabilityResp \* pBandCapability )**

Returns the band capability of the device.

**Parameters**

<i>pBand-Capability[OUT]</i>	See <a href="#">BandCapabilityResp</a> for more information of the input structure
------------------------------	--

**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

9.13.4.21 `ULONG SLQSGetBandCapability ( ULONGLONG * pBandCapability )`

Returns the band capability of the device.

## Parameters

<i>pBand-Capability</i> [OUT]	<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> <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> <li>• Bit 56 - Band class 16</li> <li>• Bit 57 - Band class 17</li> <li>• Bit 58 - Band class 18</li> <li>• Bit 59 - Band class 19</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

**9.13.4.22 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

**9.13.4.23 ULONG SLQSGetCustFeatures ( custFeaturesInfo \* pCustFeaturesInfo )**

This API fetches the current settings of custom features. This API is deprecated for EM74xx/MC74xx, please use [SLQSGetCustFeaturesV2\(\)](#) instead for EM74xx/MC74xx.

**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

**9.13.4.24 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:

- GPIOSARENABLE
- 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>
----------------------------	--

**9.13.4.25 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

#### 9.13.4.26 **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 [SLQSSwiGetSerialNoExt\(\)](#) to get MEID of CDMA modems. Timeout: 2 seconds

#### 9.13.4.27 **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. This API is deprecated for EM74xx/MC74xx, please use [SLQSSetCustFeaturesV2\(\)](#) for EM74xx/MC74xx.

##### Parameters

<i>pCustFeaturesSetting</i> [IN]	<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

#### 9.13.4.28 **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:

- GPIOARENABLE
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

## Parameters

<i>pSetCustSetting</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>See <a href="#">setCustomSettingV2</a> for more information</li> </ul>
------------------------	---

9.13.4.29 **ULONG** SLQSSwiClearDyingGaspStatistics ( )

This function Clear Dying GASP Statistics.

9.13.4.30 **ULONG** SLQSSwiGetCrashAction ( **BYTE** \* *pDevCrashState* )

This API queries the Crash State from the device.

## Parameters

<i>pDevCrash-State[OUT]</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

9.13.4.31 **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

**9.13.4.32 ULONG SLQSSwiGetDyingGaspCfg ( getDyingGaspCfg \* pConfig )**

This function queries Dying GASP Config.

**Parameters**

<i>pGetCustom-FeatureV2</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getDyingGaspCfg</a> for more information of the input structure *</li> </ul>
-----------------------------	---

**9.13.4.33 ULONG SLQSSwiGetDyingGaspStatistics ( getDyingGaspStatistics \* pStatistics )**

This function queries Dying GASP Statistics.

**Parameters**

<i>pStatistics</i>	<ul style="list-style-type: none"> <li>• See <a href="#">getDyingGaspStatistics</a> for more information of the input structure *</li> </ul>
--------------------	--

**9.13.4.34 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

**9.13.4.35 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

**9.13.4.36 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

**9.13.4.37 ULONG SLQSSwiGetHostDevInfo ( SLQSSwiGetHostDevInfoParams \* pGetHostDevInfoParams )**

This API Get Host Information from the device.

**Parameters**

<i>pGetHostDev- InfoParams</i>	<ul style="list-style-type: none"><li>• See <a href="#">SLQSSwiGetHostDevInfoParams</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: 2 Secs

**9.13.4.38 ULONG SLQSSwiGetOSInfo ( SLQSSwiGetOSInfoParams \* pParams )**

This API queries the device operating system info configured on the modem for OMA-DM reporting

**Parameters**

<i>pParams</i>	<ul style="list-style-type: none"><li>• - See <a href="#">SLQSSwiGetOSInfoParams</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: 2 Secs

#### 9.13.4.39 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>SLQSSwiGetSerialNoExtParams</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

#### 9.13.4.40 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

#### 9.13.4.41 ULONG SLQSSwiSetCrashAction ( BYTE *crashActionParams* )

This API set the Crash Action to the device.

## Parameters

<i>crashAction-Params[IN]</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

#### 9.13.4.42 ULONG SLQSSwiSetDyingGaspCfg ( setDyingGaspCfg \* pConfig )

This function set Dying GASP Config.

## Parameters

<i>pConfig</i>	<ul style="list-style-type: none"> <li>• See <a href="#">setDyingGaspCfg</a> for more information of the input structure</li> </ul>
----------------	---

#### 9.13.4.43 ULONG SLQSSwiSetHostDevInfo ( SLQSSwiSetHostDevInfoParams \* pSetHostDevInfoParams )

This API Sets the host device info configured on the modem for OMA-DM reporting

## Parameters

<i>pSetHostDev-InfoParams</i>	<ul style="list-style-type: none"> <li>• See <a href="#">SLQSSwiSetHostDevInfoParams</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: 2 Secs

**9.13.4.44    ULONG SLQSSwiSetOSInfo ( SLQSSwiSetOSInfoParams \* *pParams* )**

This API Set OS Information to the device.

## Parameters

<i>pParams</i>	<ul style="list-style-type: none"><li>• See <a href="#">SLQSSwiSetOSInfoParams</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: 2 Secs

**9.13.4.45    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>pUSBCompConfig</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

#### 9.13.4.46 **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

#### 9.13.4.47 **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[IN]</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>Optional Parameter</li> <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>Optional Parameter</li> <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

#### 9.13.4.48 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. Please use SLQSUIMGet-Configuration instead for new firmware versions and new modules.

**Parameters**

<i>id</i> [IN]	<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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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

**9.13.4.49 ULONG UIMGetICCID ( BYTE *stringSize*, CHAR \* *pString* )**

Returns the UIM ICCID. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15-C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use [SLQSUIMReadTransparent\(\)](#) (EF ID: 3F00 2FE2) instead for new firmware versions and new modules.

**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</i> [OUT]	<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

**9.13.4.50 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 [SLQSUIMGetCardStatus\(\)](#) 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>pVerifyRetries-Left[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>pUnblock-RetriesLeft[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

#### 9.13.4.51 **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[IN]</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[IN]</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</i> [IN]	<ul style="list-style-type: none"> <li>Control key de-personalization string (maximum length of 8 characters)</li> </ul>
<i>pVerifyRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <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

#### 9.13.4.52 **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 [SLQSUIMSetPinProtection\(\)](#) for new firmware versions and new modules

**Parameters**

<i>id</i> [IN]	<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> [IN]	<ul style="list-style-type: none"> <li>Enable/disable PIN protection, 0 = Disable</li> </ul>
<i>pValue</i> [IN]	<ul style="list-style-type: none"> <li>PIN value of the PIN to be enabled/disabled</li> </ul>
<i>pVerifyRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <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>Optional parameter</li> <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

#### 9.13.4.53 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> [IN]	<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</i> [IN]	<ul style="list-style-type: none"> <li>Control key de-personalization string (maximum length of 8 characters)</li> </ul>
<i>pUnblockRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Optional parameter</li> <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

#### 9.13.4.54 ULONG UIMUnblockPIN ( ULONG id, CHAR \* pPUKValue, CHAR \* pNewValue, ULONG \* pVerifyRetriesLeft, ULONG \* pUnblockRetriesLeft )

Unblocks 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> [IN]	<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</i> [IN]	<ul style="list-style-type: none"> <li>PUK value of PIN to unblock</li> </ul>
<i>pNewValue</i> [IN]	<ul style="list-style-type: none"> <li>New PIN value of PIN to unblock</li> </ul>
<i>pVerifyRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>Optional Parameter</li> <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>Optional Parameter</li> <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

#### 9.13.4.55 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> [IN]	<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>pVerifyRetries-Left[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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>pUnblock-RetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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

**9.13.4.56 ULONG ValidateSPC ( CHAR \* pSPC )**

This function Validates Service Programming code 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**

Technology Supported: CDMA  
Device Supported: MC83x5, MC7750  
Timeout: 2 seconds

**9.14 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\\_RSPLN](#) 40
- #define [DEVICE\\_SHUTDOWN](#) 5
- #define [DEVICE\\_RESET](#) 4
- #define [DEVICE\\_OFFLINE](#) 3
- #define [FIRMWARE\\_UPDATE\\_SUCCESS](#) 0x01
- #define [FIRMWARE\\_UPDATE\\_FAIL](#) 0x01
- #define [PRI\\_UPDATE\\_FAIL](#) 0x02
- #define [FIRMWARE\\_UPGRADE\\_SUCCESS](#) 0x00
- #define [IMG\\_ID\\_LEN](#) 16
- #define [BUILD\\_ID\\_LEN](#) 100
- #define [G3K\\_FIRMWARE\\_DOWNLOAD](#) 1
- #define [SPKG\\_FIRMWARE\\_DOWNLOAD](#) 2

## 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)
- void [SLQSSetSpkgFormatRequired](#) ([BYTE](#) isneeded)
- [ULONG upgrade\\_mc77xx\\_fw](#) ([LPCSTR](#) path)
- void [eGetDeviceSeries](#) (struct [sGetDeviceSeriesResult](#) \*)
- [ULONG SLQSSwiGetAllCarrierImages](#) ([ULONG](#) \*pNumOfItems, struct [SWI\\_STRUCT\\_CarrierImage](#) \*pCarrierImages, [char](#) \*pFolderPath)
- [ULONG SLQSDownloadFirmwareToSlot](#) ([CHAR](#) \*pPath, [BYTE](#) slot\_index, [BYTE](#) force\_download)
- [ULONG SLQSGetValidFwPriCombinations](#) (struct [ImageList](#) \*pStoredImageList, [ULONG](#) \*pValidCombinationSize, struct [SWI\\_STRUCT\\_CarrierImage](#) \*pValidCombinations)
- [ULONG SLQSSetSIMBasedImageSwitching](#) (void)

### 9.14.1 Detailed Description

Firmware Management Service API function prototypes.

### 9.14.2 Macro Definition Documentation

9.14.2.1 [#define BUILD\\_ID\\_LEN](#) 100

9.14.2.2 [#define DEVICE\\_OFFLINE](#) 3

9.14.2.3 [#define DEVICE\\_RESET](#) 4

9.14.2.4 [#define DEVICE\\_SHUTDOWN](#) 5

9.14.2.5 `#define FIRMWARE_UPDATE_FAIL 0x01`

9.14.2.6 `#define FIRMWARE_UPDATE_SUCCESS 0x01`

9.14.2.7 `#define FIRMWARE_UPGRADE_SUCCESS 0x00`

9.14.2.8 `#define G3K_FIRMWARE_DOWNLOAD 1`

9.14.2.9 `#define GOBI_LISTENTRIES_MAX 2`

9.14.2.10 `#define GOBI_MBN_BUILD_ID_STR_LEN 100`

9.14.2.11 `#define GOBI_MBN_IMG_ID_STR_LEN 16`

9.14.2.12 `#define GOBI_SET_IMG_PREF_RSLEN 40`

9.14.2.13 `#define IMG_ID_LEN 16`

9.14.2.14 `#define PRI_UPDATE_FAIL 0x02`

9.14.2.15 `#define SLQSFWINFO_APPVERSION_SZ 85`

9.14.2.16 `#define SLQSFWINFO_BOOTVERSION_SZ 85`

9.14.2.17 `#define SLQSFWINFO_CARRIER_SZ 20`

9.14.2.18 `#define SLQSFWINFO_CUR_CARR_NAME 17`

9.14.2.19 `#define SLQSFWINFO_CUR_CARR_REV 13`

9.14.2.20 `#define SLQSFWINFO_MODELID_SZ 20`

9.14.2.21 `#define SLQSFWINFO_PACKAGEID_SZ 85`

9.14.2.22 `#define SLQSFWINFO_PRIVERSION_SZ 16`

9.14.2.23 `#define SLQSFWINFO_SKU_SZ 15`

9.14.2.24 `#define SPKG_FIRMWARE_DOWNLOAD 2`

### 9.14.3 Enumeration Type Documentation

#### 9.14.3.1 `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`**

### 9.14.3.2 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\_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***

#### 9.14.3.3 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***

#### 9.14.3.4 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***

#### 9.14.3.5 enum eGobiImageTech

enumeration which lists the technology supported by the image

Enumerator

***eGOBI\_IMG\_TECH\_CDMA***  
***eGOBI\_IMG\_TECH\_UMTS***

### 9.14.4 Function Documentation

#### 9.14.4.1 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

**9.14.4.2 void eGetDeviceSeries ( struct sGetDeviceSeriesResult \* )**

Name : eGetDeviceSeries

**Parameters**

<i>none</i>	
-------------	--

**Returns**

[sGetDeviceSeriesResult](#)

**Note**

Get Devie Series

**9.14.4.3 ULONG GetImagesPreference ( ULONG \* plmageListSize, struct PrefImageList \* plmageList )**

restore original alignment from stack Gets the current images preference from the device.

**Parameters**

<i>plmageListSize</i> [I-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>plmageList</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

**9.14.4.4 ULONG GetImageStore ( WORD *imageStorePathSize*, CHAR \* *plmageStorePath* )**

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</i> [OUT]	<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

**9.14.4.5 ULONG GetStoredImages ( ULONG \* *plmageListSize*, BYTE \* *plmageList* )**

restore original alignment from stack Gets the list of images stored on the device.

**Parameters**

<i>plmageListSize</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>plmageList</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**

Device Supported: MC83x5/SL9090  
Timeout: 2 seconds

#### 9.14.4.6 **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>plmageList</i> [IN]	<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</i> [IN]	<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>plmageTypes-Size</i> [IN/OUT]	<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>plmageTypes</i> [OUT]	-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

#### 9.14.4.7 **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).

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. otherwise, report FW\_PREFERENCE\_MISMATCH
10. Disable callbacks and exit.

The call to this API blocks until step 7 or 10. 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>pPath[IN]</i>	<ul style="list-style-type: none"> <li>fully qualified path to firmware image to download.</li> </ul>
<i>slot_index[IN]</i>	<ul style="list-style-type: none"> <li>slot id in the modem to store the firmware</li> </ul>
<i>force_download[IN]</i>	<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\_NONE - Firmware download/Switch success.
- eQCWWAN\_ERR\_INVALID\_ARG - The path input does not contain any image
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL - Firmware download/switch failed
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH - Download success but device ofline due to image preference mismatch ( ref. syslogs for cause )

#### See Also

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

#### Note

Timeout: NA

#### 9.14.4.8 ULONG SLQSGetBootVersionNumber ( ULONG \* bootversion )

Gets the boot loader version number

## Parameters

<i>bootversion[OUT]</i>	<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

#### 9.14.4.9 ULONG SLQSGetFirmwareInfo ( struct *qmifwinfo\_s* \* *pinfo* )

Returns firmware image information from the connected device

## Parameters

<i>pinfo[OUT]</i>	<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.

#### 9.14.4.10 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

#### 9.14.4.11 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

#### 9.14.4.12 `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

#### 9.14.4.13 `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

#### 9.14.4.14 **ULONG** SLQSGetValidFwPriCombinations ( struct ImageList \* *pStoredImageList*, **ULONG** \* *pValidCombinationSize*, struct SWI\_STRUCT\_CarrierImage \* *pValidCombinations* )

This API distills valid Firmware/PRI combinations from GetStoredImages result

**Parameters**

in	<i>pStoredImageList</i>	<ul style="list-style-type: none"> <li>image list returned from GetStoredImages</li> </ul>
in, out	<i>pValidCombinationSize</i>	<ul style="list-style-type: none"> <li>number of combination passed in and returned</li> </ul>
out	<i>pValidCombinations</i>	<ul style="list-style-type: none"> <li>valid combinations returned</li> </ul>

**Returns**

- eQCWWAN\_ERR\_INVALID\_ARG - Invalid parameters
- eQCWWAN\_ERR\_BUFFER\_SZ - No enough element to store combinations returned

**See Also**

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

#### 9.14.4.15 **BOOL** SLQSIspkgFormatRequired ( void )

Check if SPKG format download is required for SL9090/MC9090, it returns the value which was set by API [SLQSSetSpkgFormatRequired\(\)](#)

**Parameters**

<i>none</i>
-------------

**Returns**

return TRUE if required, otherwise, return FALSE

**Note**

Device Supported: MC9090/SL9090

#### 9.14.4.16 **ULONG** SLQSSetSIMBasedImageSwitching ( void )

This API is used to enable the SIM-based Image Switching. The modem will reboot automatically to take effect of the enabling

**Returns**

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

**See Also**

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

**Note**

Device Supported: EM74xx/MC74xx

**9.14.4.17 void SLQSSetSpkgFormatRequired ( BYTE *isneeded* )**

Set if SPKG format download is required for SL9090/MC9090

**Parameters**

<i>isneeded</i>	[INPUT] user inputs the firmware download method preference <ul style="list-style-type: none"> <li>• 1 - Gobi3K download method, use mbn files. This is default value</li> <li>• 2 - SPKG download method, use cwe file</li> </ul>
-----------------	--

**Returns**

None

**Note**

Device Supported: MC9090/SL9090

**9.14.4.18 ULONG SLQSSwiGetAllCarrierImages ( ULONG \* *pNumOfItems*, struct SWI\_STRUCT\_CarrierImage \* *pCarrierImages*, char \* *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{IN/OUT}</li> </ul>
<i>pCarrierImages</i> [-OUT]	<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 [IN]</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.

**9.14.4.19 ULONG SLQSupgradeFirmware9x15 ( 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</i> [IN]	<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\_NONE - Firmware download/Switch success.
- eQCWWAN\_ERR\_INVALID\_ARG - The path input does not contain any image
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL - Firmware download/switch failed
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH - Download success 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

#### 9.14.4.20 ULONG upgrade\_mc77xx\_fw ( LPCSTR path )

#### 9.14.4.21 ULONG UpgradeFirmware2k ( CHAR \* pDestinationPath )

This API is used to download firmware to a MC77xx or Gobi 3000 device. For SL909/MC9090, [SLQSSetSpkgFormatRequired\(\)](#) needs to be called in advance to specify the download method, Gobi3K or SPKG download, please refer to the API [SLQSSetSpkgFormatRequired\(\)](#) for more details of the input values. If [SLQSSetSpkgFormatRequired\(\)](#) is not called in advance, it will use Gobi3K firmware download method (MBN files) as the default download method.

This API Performs the following steps:

1. Verifies arguments.
2. Verify that device Crash State should be 1 (RESET State).
3. Informs the SDK of the firmware upgrade path
4. Updates the images preference on the currently connected device.
5. 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>pDestinationPath</i> [IN]	<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. "<a href="#">&lt;path&gt;\to\carrier\image&gt;\&lt;carrier index&gt;\</a>" where <a href="#">&lt;carrier index&gt;</a>="" is a valid sub-directory entry. For 9x30 devices if <i>pDestinationPath</i> 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

## 9.15 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](#) \*pImConfigIndRegisterInfo)

### 9.15.1 Detailed Description

IMS Service API function prototypes.

### 9.15.2 Function Documentation

#### 9.15.2.1 [ULONG SLQSGetIMSSMSConfig](#) ( [GetIMSSMSConfigParams](#) \* *pGetIMSSMSConfigParams* )

This API retrieves the SMS configuration parameters.

##### Parameters

<i>pGetIMSSMSConfigParams</i> [I-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: MC73xx, MC74xx and EM74xx  
Timeout: 5 seconds

**9.15.2.2 ULONG SLQSGetIMSUserConfig ( GetIMSUserConfigParams \* pGetIMSUserConfigParams )**

This API retrieves the IMS User configuration parameters.

**Parameters**

<i>pGetIMSUser- ConfigParams[IN/ OUT]</i>	<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: MC73xx, MC74xx and EM74xx  
Timeout: 5 seconds

**9.15.2.3 ULONG SLQSGetIMSVoIPConfig ( GetIMSVoIPConfigResp \* pGetIMSVoIPConfigResp )**

This API retrieves the IMS VoIP configuration parameters.

**Parameters**

<i>GetIMSVoIP- ConfigResp[OUT]</i>	<ul style="list-style-type: none"><li>• See <a href="#">GetIMSVoIPConfigResp</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, MC74xx and EM74xx  
Timeout: 5 seconds

**9.15.2.4 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 GetRegMgrConfigResp 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, MC74xx and EM74xx  
Timeout: 5 seconds

**9.15.2.5 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: MC73xx, MC74xx and EM74xx  
Timeout: 5 seconds

### 9.15.2.6 **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: MC73xx, MC74xx and EM74xx  
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.

### 9.15.2.7 **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: MC73xx, MC74xx and EM74xx  
Timeout: 5 seconds

### 9.15.2.8 **ULONG** SLQSSetIMSUserConfig ( **SetIMSUserConfigReq** \* *pSetIMSUserConfigReq*, **SetIMSUserConfigResp** \* *pSetIMSUserConfigResp* )

This API sets the IMS user configuration parameters for the requesting control point.

#### Parameters

<i>pSetIMSUserConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSUserConfigReq</a> for more information</li> </ul>
<i>pSetIMSUserConfigResp</i> [OUT]	<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: MC73xx, MC74xx and EM74xx  
 Timeout: 5 seconds

### 9.15.2.9 **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</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetIMSVoIPConfigReq</a> for more information</li> </ul>
<i>pSetIMSVoIPConfigResp</i> [OUT]	<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: MC73xx, MC74xx and EM74xx  
 Timeout: 5 seconds

#### 9.15.2.10 **ULONG** SLQSSetRegMgrConfig ( **SetRegMgrConfigReq** \* *pSetRegMgrConfigReq*, **SetRegMgrConfigResp** \* *pSetRegMgrConfigResp* )

This API sets the IMS registration manager configuration parameters for the requesting control point.

##### Parameters

<i>pSetRegMgrConfigReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SetRegMgrConfigReq</a> for more information</li> </ul>
<i>pSetRegMgrConfigResp</i> [OUT]	<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: MC73xx, MC74xx and EM74xx  
 Timeout: 5 seconds

#### 9.15.2.11 **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: MC73xx, MC74xx and EM74xx  
 Timeout: 5 seconds

## 9.16 qaGobiApilmsa.h File Reference

IMS A Service API function prototypes.

### Data Structures

- struct [IMS AIndRegisterInfo](#)
- struct [SupportedMsgList](#)
- struct [IMSASupportedMsgInfo](#)
- struct [ReqFieldsList](#)
- struct [RespFieldsList](#)
- struct [IndFieldsList](#)
- struct [IMSASupportedFieldsResp](#)
- struct [IMS ARegistrationStatus](#)
- struct [IMS AServiceStatus](#)

### Functions

- [ULONG SLQSRegisterIMS AIndication](#) ([IMS AIndRegisterInfo](#) \*pImsaIndRegisterInfo)
- [ULONG SLQSGetIMSASupportedMsg](#) ([IMSASupportedMsgInfo](#) \*pIMSASupportedMsgInfo)
- [ULONG SLQSGetIMSASupportedFields](#) ([WORD](#) messageID, [IMSASupportedFieldsResp](#) \*pIMSASupportedFieldsResp)
- [ULONG SLQSGetIMS ARegStatus](#) ([IMS ARegistrationStatus](#) \*pIMS ARegistrationStatus)
- [ULONG SLQSGetIMS AServiceStatus](#) ([IMS AServiceStatus](#) \*pIMS AServiceStatus)

#### 9.16.1 Detailed Description

IMS A Service API function prototypes.

#### 9.16.2 Function Documentation

##### 9.16.2.1 [ULONG SLQSGetIMS ARegStatus](#) ( [IMS ARegistrationStatus](#) \* *pIMS ARegistrationStatus* )

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

##### Parameters

<i>pIMS ARegistrationStatus</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="#">IMS ARegistrationStatus</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.

### 9.16.2.2 **ULONG SLQSGetIMSAServiceStatus ( IMSAServiceStatus \* *pIMSAServiceStatus* )**

Gets the service status for various IMS services for the requesting control point.

**Parameters**

<i>pIMSAServiceStatus</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.

### 9.16.2.3 **ULONG SLQSGetIMSASupportedFields ( WORD *messageID*, IMSASupportedFieldsResp \* *pIMSASupportedFieldsResp* )**

Queries the set of supported fields implemented by the currently running software.

**Parameters**

<i>messageID</i> [IN]	<ul style="list-style-type: none"> <li>Service Message ID.</li> </ul>
<i>pIMSASupportedFieldsResp</i> [OUT]	<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.

#### 9.16.2.4 ULONG SLQSGetIMSASupportedMsg ( IMSASupportedMsgInfo \* *pIMSASupportedMsgInfo* )

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

**Parameters**

<i>pIMSASupportedMsgInfo</i> [OUT]	<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

#### 9.16.2.5 ULONG SLQSRegisterIMSAIndication ( IMSAIndRegisterInfo \* *plmsaIndRegisterInfo* )

Sets the registration state for different QMI\_IMSA indications for the requesting control point

**Parameters**

<i>plmsaIndRegisterInfo</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.

## 9.17 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 [CellDb](#)
- struct [CikInfo](#)
- struct [BdsSV](#)
- struct [BdsSVInfo](#)
- struct [LocDelAssDataReq](#)
- struct [SwiLocGetAutoStartResp](#)
- struct [SwiLocSetAutoStartReq](#)
- struct [altitudeSrcInfo](#)
- struct [LocInjectPositionReq](#)
- struct [LocSetCradleMountReq](#)
- struct [sensorData](#)
- struct [tempratureData](#)
- struct [LocInjectSensorDataReq](#)

### Macros

- [#define MAX\\_SENSOR\\_DATA\\_LEN 64](#)
- [#define MAX\\_TEMP\\_DATA\\_LEN 64](#)

### 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)
- [ULONG SwiLocGetAutoStart](#) ([SwiLocGetAutoStartResp](#) \*resp)
- [ULONG SwiLocSetAutoStart](#) ([SwiLocSetAutoStartReq](#) \*req)
- [ULONG SLQSLOCInjectUTCTime](#) ([ULONGLONG](#) timeMsec, [ULONG](#) timeUncMsec)

- [ULONG SLQSLOCInjectPosition](#) ([LocInjectPositionReq](#) \*pLocInjectPositionReq)
- [ULONG SLQSLOCSetCradleMountConfig](#) ([LocSetCradleMountReq](#) \*pLocSetCradleMountReq)
- [ULONG SLQSLOCInjectSensorData](#) ([LocInjectSensorDataReq](#) \*pLocInjectSensorDataReq)
- [ULONG SLQSLOCGetBestAvailPos](#) ([ULONG](#) xid)

### 9.17.1 Detailed Description

Location API function prototypes.

### 9.17.2 Macro Definition Documentation

9.17.2.1 `#define MAX_SENSOR_DATA_LEN 64`

9.17.2.2 `#define MAX_TEMP_DATA_LEN 64`

### 9.17.3 Function Documentation

9.17.3.1 `ULONG SLQSLOCDeAssData ( LocDeAssDataReq request )`

Used by the control point to delete the location engine assistance data

#### Parameters

<i>request[IN]</i>	<ul style="list-style-type: none"> <li>• request structure parameters should contain all NULL pointers 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

9.17.3.2 `ULONG SLQSLOCEventRegister ( LOCEventRegisterReqResp * pLOCEventRegisterReqResp )`

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

#### Parameters

<i>pLOCEvent-RegisterReq-Resp[IN]</i>	<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

**9.17.3.3 ULONG SLQSLOCGetBestAvailPos ( ULONG xid )**

Control point to get the best available position estimate from the location engine.

**Parameters**

<i>xid</i> [IN]	<ul style="list-style-type: none"><li>• Identifies the transaction.</li><li>• The transaction ID is returned in the Get Best Available Position indication.</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

- GPS engine should be started to get best available position.

**9.17.3.4 ULONG SLQSLOCInjectPosition ( LocInjectPositionReq \* pLocInjectPositionReq )**

Injects a position to the location engine.

**Parameters**

<i>pLocInject-PositionReq</i> [IN]	<ul style="list-style-type: none"><li>• See <a href="#">LocInjectPositionReq</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

### 9.17.3.5 ULONG SLQSLOCInjectSensorData ( LocInjectSensorDataReq \* pLocInjectSensorDataReq )

Control point to to inject sensor data into the GNSS location engine.

#### Parameters

<i>pLocInjectSensorDataReq</i> [IN]	<ul style="list-style-type: none"><li>See <a href="#">LocInjectSensorDataReq</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

### 9.17.3.6 ULONG SLQSLOCInjectUTCtime ( ULONGLONG timeMsec, ULONG timeUncMsec )

Injects UTC time in the location engine.

#### Parameters

<i>timeMsec</i> [IN]	<ul style="list-style-type: none"><li>The UTC time since Jan. 1, 1970</li></ul>
<i>timeUncMsec</i> [IN]	<ul style="list-style-type: none"><li>The time Uncertainty</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

### 9.17.3.7 ULONG SLQSLOCSetCradleMountConfig ( LocSetCradleMountReq \* pLocSetCradleMountReq )

Control point to set the current cradle mount configuration.

#### Parameters

<i>pLocSetCradleMountReq</i> [IN]	<ul style="list-style-type: none"><li>See <a href="#">LocSetCradleMountReq</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

### 9.17.3.8 **ULONG SLQSLCSetExtPowerState ( LOCExtPowerStateReqResp \* pLOCExtPowerStateReqResp )**

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

**Parameters**

<i>pLOCExtPowerStateReqResp</i> [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

### 9.17.3.9 **ULONG SLQSLCSetOpMode ( 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> [IN]	<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

**9.17.3.10 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

**9.17.3.11 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

### 9.17.3.12 **ULONG** SwiLocGetAutoStart ( **SwiLocGetAutoStartResp** \* *resp* )

Used by the control point to Get Loc Auto Start settings

#### Parameters

<i>resp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">SwiLocGetAutoStartResp</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

### 9.17.3.13 **ULONG** SwiLocSetAutoStart ( **SwiLocSetAutoStartReq** \* *req* )

Used by the control point to Set Loc Auto Start settings

#### Parameters

<i>req</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">SwiLocSetAutoStartReq</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

## 9.18 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 [LTESSInfo](#)
- struct [TDSCDMASigInfoExt](#)
- struct [nasGetSigInfoResp](#)
- struct [nasIndicationRegisterReq](#)
- struct [nasPLMNNNameReq](#)
- struct [nasPLMNNNameResp](#)
- 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 [IOThresh](#)
- struct [RSRQThresh](#)
- struct [RSRPThresh](#)
- struct [LTESigRptCfg](#)
- struct [TDSCDMASINRCONFTthresh](#)
- struct [sigInfo](#)
- struct [NasSwiIndReg](#)
- struct [CDMARSSIThresh](#)
- struct [CDMAECIOThresh](#)
- struct [HRRSSIThresh](#)

- struct [HDRECIOTresh](#)
- struct [HDSINRThreshold](#)
- struct [HDRIOTresh](#)
- struct [GSMRSSITresh](#)
- struct [WCDMARSSITresh](#)
- struct [WCDMAECIOTresh](#)
- struct [LTERSSITresh](#)
- struct [LTERSNRThreshold](#)
- struct [LTERSRQTresh](#)
- struct [LTERSRPTresh](#)
- struct [LTERSigRptConfig](#)
- struct [TDSCDMARSCPTresh](#)
- struct [TDSCDMARSSITresh](#)
- struct [TDSCDMAECIOTresh](#)
- struct [TDSCDMASINRTresh](#)
- struct [setSignalStrengthInfo](#)
- struct [PhyCaAggScellIndType](#)
- struct [PhyCaAggScellIDBw](#)
- struct [PhyCaAggScellInfo](#)
- struct [PhyCaAggPcellInfo](#)
- struct [PhyCaAggScellIndex](#)
- struct [nasGetLTECphyCaResp](#)
- struct [nasGetLTECphyCa](#)
- struct [wcdmaUARFCN](#)
- struct [lteEARFCN](#)
- struct [ltePCI](#)
- struct [nasSwiGetChannelLockResp](#)
- struct [nasSwiSetChannelLockReq](#)
- struct [timeInfo](#)
- struct [GetNetworkTimeResp](#)

## 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
- #define [NAS\\_SIG\\_INFO\\_MAX\\_TDSCDMA\\_THRESHOLDS\\_LIST\\_SIZE](#) 16
- #define [NAS\\_SIG\\_INFO\\_MIN\\_dBm\\_FLOAT\\_VALUE](#) -125.0
- #define [NAS\\_SIG\\_INFO\\_MIN\\_dB\\_FLOAT\\_VALUE](#) -10.0

## Typedefs

- typedef struct [\\_SlqsNas3GppNetworkRAT](#) SlqsNas3GppNetworkRAT
- 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 }
- enum `eSYS_SRV_DOMAIN` {  
`eSYS_SRV_DOMAIN_NO_SRV` = 0x00,  
`eSYS_SRV_DOMAIN_CS_ONLY` = 0x01,  
`eSYS_SRV_DOMAIN_PS_ONLY` = 0x02,  
`eSYS_SRV_DOMAIN_CS_PS` = 0x03,  
`eSYS_SRV_DOMAIN_CAMPED` = 0x04,  
`eSYS_SRV_DOMAIN_UNKNOWN` }

## 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 GetANAAAuthenticationStatus (ULONG *pStatus)`
- `ULONG SLQSGetServingSystem (qaQmiServingSystemParam *pServingSystem)`
- `ULONG SLQSSetBandPreference (ULONGLONG 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 SLQSNasInitiateNetworkRegistration](#) ([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 SLQSNasSwiIndicationRegister](#) ([NasSwiIndReg](#) \*pIndRegReq)
- [ULONG GetHomeNetwork3GPP2](#) ([WORD](#) \*pMCC, [WORD](#) \*pMNC, [BYTE](#) nameSize, [CHAR](#) \*pName, [WORD](#) \*pSID, [WORD](#) \*pNID, [WORD](#) \*pNw2MCC, [WORD](#) \*pNw2MNC, [BYTE](#) \*pNw2DescDisp, [BYTE](#) \*pNw2DescEnc, [BYTE](#) \*pNw2DescLen, [BYTE](#) \*pNw2Name)
- [ULONG SLQSNasConfigSigInfo2](#) ([setSignalStrengthInfo](#) \*pSetSignalStrengthInfo)
- [ULONG SLQSNASGetLTECPHYCaInfo](#) ([nasGetLTECphyCa](#) \*pLTECPhyCa)
- [ULONG SLQSNasIndicationRegisterLTECphyCa](#) ([BYTE](#) \*bStatus)
- [ULONG SLQSNASSwiGetChannelLock](#) ([nasSwiGetChannelLockResp](#) \*pNasSwiGetChannelLockResp)
- [ULONG SLQSNASSwiSetChannelLock](#) ([nasSwiSetChannelLockReq](#) \*pNasSwiSetChannelLockReq)
- [ULONG SLQSGetNetworkTime](#) ([GetNetworkTimeResp](#) \*pGetNetworkTimeResp)

### 9.18.1 Detailed Description

Network Access Service API function prototypes.

### 9.18.2 Macro Definition Documentation

9.18.2.1 `#define IMSI_M_S1_LENGTH 7`

9.18.2.2 `#define IMSI_M_S2_LENGTH 3`

9.18.2.3 `#define MAX_DATA_SRV_CAPABILITIES 0x20`

9.18.2.4 `#define MAX_DESCRIPTION_LENGTH 255`

9.18.2.5 `#define MAX_PILOT_SETS 0xFF`

9.18.2.6 `#define MAX_SERV_SYSTEM_RADIO_INTERFACES 0x0A`

9.18.2.7 `#define NAM_NAME_LENGTH 12`

9.18.2.8 `#define NAS_SIG_INFO_MAX_TDSCDMA_THRESHOLDS_LIST_SIZE 16`

9.18.2.9 `#define NAS_SIG_INFO_MIN_dB_FLOAT_VALUE -10.0`

9.18.2.10 `#define NAS_SIG_INFO_MIN_dBm_FLOAT_VALUE -125.0`

9.18.2.11 `#define PLMN_LENGTH 3`

9.18.2.12 `#define SLQS_SS_INFO_LIST_MAX_ELEMENTS 18`

9.18.2.13 `#define SLQS_SYSTEM_ID_SIZE 16`

9.18.2.14 `#define UATISIZE 16`

### 9.18.3 Typedef Documentation

9.18.3.1 `typedef struct _SlqsNas3GppNetworkRAT_ SlqsNas3GppNetworkRAT`

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> <li>– 0x08 - LTE</li> <li>– 0x09 - TD-SCDMA</li> </ul> </li> </ul>

9.18.3.2 `typedef struct _slqsNetworkScanInfo slqsNetworkScanInfo`

Contain the network scan information.

#### Parameters

<i>pNetworkInfo-Instances[IN/O-UT]</i>	<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[O-UT]</i>	<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[IN/OUT]</i>	<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[OUT]</i>	<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>pPCSDigit-Instances[IN/OUT]</i>	<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[OUT]</i>	<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[OUT]</i>	<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>

#### 9.18.3.3 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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.</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 22 - E-UTRA Operating Band 23</li> <li>– Bit 23 - E-UTRA Operating Band 24</li> <li>– Bit 24 - E-UTRA Operating Band 25</li> <li>– Bit 25 - E-UTRA Operating Band 26</li> <li>– Bit 27 - E-UTRA Operating Band 28</li> <li>– Bit 28 - E-UTRA Operating Band 29</li> <li>– Bit 29 - E-UTRA Operating Band 32</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> <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> <li>– Bit 40 - E-UTRA Operating Band 41</li> <li>– Bit 41 - E-UTRA Operating Band 42</li> <li>– Bit 42 - E-UTRA Operating Band 43</li> <li>– Bit 60 - E-UTRA Operating Band 125</li> <li>– All other bits are reserved</li> </ul> </li> <li>• function <a href="#">SLQSGetSysSelectionPref()</a> returns a default value FFFFFFFFFFFFFFFF if no value is returned by the device.</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

**9.18.3.4 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> <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> <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>– Bit 32 to 47 - Reserved</li> <li>– Bit 48 - WCDMA Europe 2600 band</li> <li>– Bit 49 - WCDMA Europe and Japan 900 band</li> <li>– Bit 50 - WCDMA Japan 1700 band</li> <li>– Bit 51 to 55 - Reserved</li> <li>– Bit 56 - Band Class 16</li> <li>– Bit 57 - Band Class 17</li> <li>– Bit 58 - Band Class 18</li> <li>– Bit 59 - Band Class 19</li> <li>– Bit 60 to 64 - Reserved</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 22 - E-UTRA Operating Band 23</li> <li>– Bit 23 - E-UTRA Operating Band 24</li> <li>– Bit 24 - E-UTRA Operating Band 25</li> <li>– Bit 25 - E-UTRA Operating Band 26</li> <li>– Bit 27 - E-UTRA Operating Band 28</li> <li>– Bit 28 - E-UTRA Operating Band 29</li> <li>– Bit 29 - E-UTRA Operating Band 32</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> <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> <li>– Bit 40 - E-UTRA Operating Band 41</li> <li>– Bit 41 - E-UTRA Operating Band 42</li> <li>– Bit 42 - E-UTRA Operating Band 43</li> <li>– Bit 60 - E-UTRA Operating Band 125</li> <li>– All other bits are reserved</li> </ul> </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>• Either of pNetSelPref or pCSGID can be set.</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>• At least one system selection setting to be set if pChgDuration is populated.</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>• pNetSelPref is expected if MNC includes PCS digit is set to 1.</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>pTdsdmaBand-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating bitmask representing the TD-SCDMA band preference to be set.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - TD-SCDMA Band A</li> <li>– 0x02 - TD-SCDMA Band B</li> <li>– 0x04 - TD-SCDMA Band C</li> <li>– 0x08 - TD-SCDMA Band D</li> <li>– 0x10 - TD-SCDMA Band E</li> <li>– 0x20 - TD-SCDMA Band F</li> <li>– All other bits are reserved</li> </ul> </li> </ul>

<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> <li>- <a href="#">acqOrderPref</a></li> <li>• Optional parameter for specifying Acquisition Order Preference</li> <li>• see <a href="#">acqOrderPref</a> for more information</li> </ul>
<i>pSrvReg-Restriction</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Network Selection Registration Restriction Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Device follows the normal registration process</li> <li>– 0x01 - Device camps on the network according to its provisioning, but does not register</li> <li>– 0x02 - Device selects the network for limited service</li> <li>– All other values are reserved.</li> </ul> </li> </ul>
<i>pCSGID</i>	<ul style="list-style-type: none"> <li>- <a href="#">CSGID</a></li> <li>• Optional parameter for specifying CSG ID</li> <li>• Either of pNetSelPref or pCSGID can be set.</li> <li>• see <a href="#">CSGID</a> for more information</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>

### 9.18.4 Enumeration Type Documentation

#### 9.18.4.1 enum \_NAMS\_RADIO\_IF\_TECHNOLOGY\_

Enumerator

***eNAS\_RADIO\_IF\_GSM***

***eNAS\_RADIO\_IF\_UMTS***

***eNAS\_RADIO\_IF\_LTE***

***eNAS\_RADIO\_IF\_TDSCDMA***

#### 9.18.4.2 enum eSYS\_SRV\_DOMAIN

Enumerator

***eSYS\_SRV\_DOMAIN\_NO\_SRV***

***eSYS\_SRV\_DOMAIN\_CS\_ONLY***

***eSYS\_SRV\_DOMAIN\_PS\_ONLY***

***eSYS\_SRV\_DOMAIN\_CS\_PS***

***eSYS\_SRV\_DOMAIN\_CAMPED***

***eSYS\_SRV\_DOMAIN\_UNKNOWN***

## 9.18.4.3 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***

## 9.18.4.4 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***

## 9.18.5 Function Documentation

## 9.18.5.1 ULONG GetACCOLC ( BYTE \* pACCOLC )

Retrieves information about the access overload class (ACCOLC)

## Parameters

<i>pACCOLC</i> [OUT]	<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

## 9.18.5.2 ULONG GetANAAAAuthenticationStatus ( ULONG \* pStatus )

AN-AAA authentication status of the device.

## Parameters

<i>pStatus</i> [OUT]	<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> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>– 2 - Not Requested</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

**9.18.5.3** **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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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[OUT]</i>	<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[OUT]</i>	<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[OUT]</i>	<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[OUT]</i>	<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

#### 9.18.5.4 **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</i> [OUT]	<ul style="list-style-type: none"> <li>Mobile country code (UMTS only).</li> </ul>
<i>pMNC</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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

#### 9.18.5.5 **ULONG** GetHomeNetwork3GPP2 ( **WORD** \* *pMCC*, **WORD** \* *pMNC*, **BYTE** *nameSize*, **CHAR** \* *pName*, **WORD** \* *pSID*, **WORD** \* *pNID*, **WORD** \* *pNw2MCC*, **WORD** \* *pNw2MNC*, **BYTE** \* *pNw2DescDisp*, **BYTE** \* *pNw2DescEnc*, **BYTE** \* *pNw2DescLen*, **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

#### 9.18.5.6 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 [SLQSGetSysSelectionPref\(\)](#) for new firmware versions and new modules

## Parameters

<i>pTechnologyPref</i> [OUT]	<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</i> [OUT]	<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</i> [OUT]	<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

### 9.18.5.7 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

### 9.18.5.8 ULONG GetServingNetwork ( ULONG \* *pRegistrationState*, ULONG \* *pCSDomain*, ULONG \* *pPSDomain*, ULONG \* *pRAN*, BYTE \* *pRadiofacesSize*, BYTE \* *pRadiofaces*, 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>pRegistrationState</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</i> [OUT]	<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>pRadioIfaceSize[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>pRadioIface[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

#### 9.18.5.9 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[IN/OUT]</i>	<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[OUT]</i>	<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

#### 9.18.5.10 **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[IN/OUT]</i>	<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>pSignalStrength[OUT]</i>	<ul style="list-style-type: none"> <li>• Received signal strength array (in dBm)</li> </ul>
<i>pRadioInterface[OUT]</i>	<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

**9.18.5.11 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[IN]</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

**9.18.5.12 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

**9.18.5.13 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</i> [OUT]	<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

**9.18.5.14 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[IN]</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

#### 9.18.5.15 **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[IN]</i>	<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[IN]</i>	<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[IN]</i>	<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> [IN]	<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

#### 9.18.5.16 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> [ <i>N</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> [ <i>N</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

## 9.18.5.17 ULONG SLQSCfgSigInfo ( 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. Please use [SLQSNasConfigSigInfo2\(\)](#) instead for new firmware versions and new modules.

## Parameters

<i>pSigInfo</i> [ <i>N</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

#### 9.18.5.18 **ULONG SLQSGetErrorRate ( GetErrRateResp \* pGetErrRateResp )**

This API retrieves current error rate information

**Parameters**

<i>pGetErrRateResp</i> [OUT]	<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

#### 9.18.5.19 **ULONG SLQSGetNetworkTime ( GetNetworkTimeResp \* pGetNetworkTimeResp )**

This API retrieves the last known network time information from the UE.

**Parameters**

<i>pGetNetworkTimeResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">GetNetworkTimeResp</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

#### 9.18.5.20 **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

#### 9.18.5.21 **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

#### 9.18.5.22 **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

#### 9.18.5.23 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

#### 9.18.5.24 ULONG SLQSGetSysSelectionPref ( sysSelectPrefInfo \* pSysSelectPrefInfo )

Queries the different system selection preferences of the device.

**Parameters**

<i>pSysSelectPrefInfo</i> [OUT]	<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

**9.18.5.25 ULONG SLQSIInitiateNetworkRegistration ( nasInitNetworkReq \* pNasInitNetRegistrationReg )**

Initiates the network registration process.

**Parameters**

<i>pNasInitNet-Registration-Req[IN]</i>	<ul style="list-style-type: none"> <li>• Pointer to structure nasInitNetworkReq               <ul style="list-style-type: none"> <li>– See <a href="#">nasInitNetworkReq</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

**9.18.5.26 ULONG SLQSNasConfigSigInfo2 ( setSignalStrengthInfo \* pSetSignalStrengthInfo )**

Sets the signal strength reporting thresholds

**Parameters**

<i>pSetSignalStrengthInfo[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.

**9.18.5.27** **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.

**9.18.5.28** **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.

#### 9.18.5.29 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

#### 9.18.5.30 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.

#### 9.18.5.31 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

**9.18.5.32 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.

**9.18.5.33 ULONG SLQSNasGetTxRxInfo ( nasGetTxRxInfoReq \* pGetTxRxInfoReq, nasGetTxRxInfoResp \* pGetTxRxInfoResp )**

This API retrieves the detailed Tx/Rx information.

**Parameters**

<i>pGetTxRxInfoReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">nasGetTxRxInfoReq</a> for more information</li> </ul>
<i>pGetTxRxInfoResp</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.

#### 9.18.5.34 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> [IN]	<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> [IN]	<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> [IN]	<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

**9.18.5.35    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**

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.

**9.18.5.36    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.

**9.18.5.37    ULONG SLQSNASSwiGetChannelLock ( nasSwiGetChannelLockResp \* pNasSwiGetChannelLockResp )**

This API queries the channel or cell which the UE is locked into.

## Parameters

<i>pNasSwiGetChannelLock[OUT]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nasSwiGetChannelLockResp</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.

#### 9.18.5.38 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

#### 9.18.5.39 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

#### 9.18.5.40 **ULONG** SLQSNASSwiSetChannelLock ( **nasSwiSetChannelLockReq** \* *pNasSwiSetChannelLockReq* )

This API allows the host to lock the UE to a specific channel or cell.

**Parameters**

<i>pNasSwiSetChannelLockReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">nasSwiSetChannelLockReq</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**

The settings are persistent across reboots.

#### 9.18.5.41 **ULONG** SLQSPerformNetworkScan ( **slqsNetworkScanInfo** \* *pNetworkInfo* )

Performs scan for available networks and scans for RAT info as well.

**Parameters**

<i>pNetworkInfo</i> [I- N/OUT]	<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

9.18.5.42 **ULONG** SLQSSetBandPreference ( **ULONGLONG** *bandpreference* )

Provides information about the band preference.

## Parameters

<i>bandpreference</i> [ IN]	<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> <li>– 59 - BC19 - Band Class 19</li> <li>– 60 to 64 - Reserved</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

**9.18.5.43 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

**9.18.5.44 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

#### 9.18.5.45 **ULONG** SLQSSwiGetHDRProtSubtype ( HDRProtSubtypResp \* pHDRProtSubtypResp )

This API retrieves HDR Prototype Subtype related information

## Parameters

<i>pHDRProtSubtypResp</i> [OUT]	<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

#### 9.18.5.46 **ULONG** SLQSSwiGetHRPDStats ( GetHRPDStatsResp \* pGetHRPDStatsResp )

This API retrieves currently acquired HRPD system statistics

## Parameters

<i>pGetHRPDStatsResp</i> [OUT]	<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

#### 9.18.5.47 **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

#### 9.18.5.48 ULONG SLQSSwiNetworkDebug ( NetworkDebugResp \* pNetworkDebugResp )

This API retrieves device and network status details

## Parameters

<i>pNetworkDebugResp[OUT]</i>	<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

#### 9.18.5.49 ULONG SLQSSwiPSDetach ( PSDetachReq \* pPSDetachReq )

This API detaches PS connection.

## Parameters

<i>pPSDetachReq[IN]</i>	<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

## 9.19 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)

### 9.19.1 Detailed Description

Open Mobile Alliance Device Management Service API function prototypes.

### 9.19.2 Function Documentation

#### 9.19.2.1 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

#### 9.19.2.2 ULONG OMADMGetPendingNIA ( ULONG \* pSessionType, USHORT \* pSessionID )

Returns information about the pending network-initiated alert

## Parameters

<i>SessionType</i> [OUT]	<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</i> [OUT]	<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

**9.19.2.3** **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</i> [OUT]	<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> [OUT]	<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[OUT]</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>
<i>RetryCount[OUT]</i>	<ul style="list-style-type: none"> <li>• Session retry count (when state indicates retrying)</li> </ul>
<i>SessionPause[OUT]</i>	<ul style="list-style-type: none"> <li>• Session pause timer (in seconds , when state indicates retrying)</li> </ul>
<i>Time-Remaining[OUT]</i>	<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

**9.19.2.4 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

**9.20 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](#) [GetPDSDDefaults](#) ([ULONG](#) \*pOperation, [BYTE](#) \*pTimeout, [ULONG](#) \*pInterval, [ULONG](#) \*pAccuracy)
- [ULONG](#) [SetPDSDDefaults](#) ([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](#) [ResetPDSDData](#) ([ULONG](#) \*pGPSDataMask, [ULONG](#) \*pCellDataMask)
- [ULONG](#) [SLQSSetAGPSCfg](#) ([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)

### 9.20.1 Detailed Description

Position Determination Service API function prototypes.

### 9.20.2 Macro Definition Documentation

9.20.2.1 `#define DEFAULTBYTEVALUE 0xFF`

9.20.2.2 `#define DEFAULTLONGVALUE 0xFFFFFFFF`

9.20.2.3 `#define DEFAULTWORDVALUE 0xFFFF`

### 9.20.3 Enumeration Type Documentation

9.20.3.1 anonymous enum

Enumerator

***eSetServiceAutomaticTrackingDisable***

***eSetServiceAutomaticTrackingEnable***

### 9.20.4 Function Documentation

9.20.4.1 **`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

9.20.4.2 **`ULONG GetPDSDetails ( 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

#### 9.20.4.3 ULONG GetPDSState ( ULONG \* *pEnabledStatus*, ULONG \* *pTrackingStatus* )

Returns the current PDS state.

## Parameters

<i>pEnabledStatus</i> [OUT]	<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>pTrackingStatus</i> [OUT]	<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

#### 9.20.4.4 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

#### 9.20.4.5 ULONG GetServiceAutomaticTracking ( ULONG \* pbAuto )

Returns the automatic tracking state for the service.

##### Parameters

<i>pbAuto[OUT]</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

#### 9.20.4.6 **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

#### 9.20.4.7 **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

#### 9.20.4.8 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

#### 9.20.4.9 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>systemDiscontinuities</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

#### 9.20.4.10 ULONG ResetPDSData ( ULONG \* *pGPSDataMask*, ULONG \* *pCellDataMask* )

Resets the specified PDS data.

## Parameters

<i>pGPSDataMask</i> [IN]	<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

**9.20.4.11    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

**9.20.4.12    ULONG SetPDSState (    ULONG *enable* )**

Sets the PDS state.

**Parameters**

<i>enable</i> [IN]	<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

**9.20.4.13 ULONG SetPortAutomaticTracking ( ULONG bAuto )**

Sets the automatic tracking configuration for the NMEA COM port.

**Parameters**

<i>bAuto</i> [IN]	<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

**9.20.4.14 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> [IN]	<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

**9.20.4.15    ULONG SetXTRAAutomaticDownload (    ULONG *bEnabled*,    USHORT *interval* )**

Sets the XTRA automatic database download configuration.

## Parameters

<i>bEnabled</i> [IN]	<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> [IN]	<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

**9.20.4.16    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> [IN]	<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

**9.20.4.17** `ULONG SLQSGetAGPSConfig ( ULONG * pServerAddress, ULONG * pServerPort, BYTE * pServerURL, BYTE * pServerURLLength, BYTE * pNetworkMode )`

Gets the PDS AGPS (MS-based) configuration.

## Parameters

<i>pServerAddress</i> [OUT]	<ul style="list-style-type: none"> <li>IPv4 address of AGPS server. "0" if not set</li> </ul>
<i>pServerPort</i> [OUT]	<ul style="list-style-type: none"> <li>Port number of AGPS server. "0" if not set</li> </ul>
<i>pServerURL</i> [OUT]	<ul style="list-style-type: none"> <li>URL of the AGPS server. "0" if not set</li> </ul>
<i>pServerURLLength</i> [OUT]	<ul style="list-style-type: none"> <li>URL length of AGPS server. "0" if not set</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

**9.20.4.18** `ULONG SLQSGetGPSStateInfo ( GPSStateInfo * pGPSStateInfo )`

Queries the MSM GPS server for receiver state information

## Parameters

<i>pGPSStateInfo</i> [OUT]	<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

**9.20.4.19 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

**9.20.4.20 ULONG SLQSPDSInjectAbsoluteTimeReference ( ULONGLONG timeMsec, ULONG timeUncMsec, BYTE timeBase, BYTE forceFlag )**

Injects a absolute time reference into the PDS engine.

**Parameters**

<i>timeMsec</i> [IN]	<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> [IN]	<ul style="list-style-type: none"> <li>Time uncertainty in milliseconds</li> </ul>
<i>timeBase</i> [IN]	<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> [IN]	<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

#### 9.20.4.21 **ULONG** SLQSPDSInjectPositionData ( **struct** PDSPositionData \* *pPositionData* )

Injects position data into the PDS engine.

**Parameters**

<i>pPositionData</i> [I/-N]	<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

#### 9.20.4.22 **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

#### 9.20.4.23 ULONG SLQSSetPositionMethodState ( PDSPosMethodStateReq \* pPDSPosMethodStateReq )

Sets the state of positioning methods for the device.

**Parameters**

<i>pPDSPosMethodStateReq</i> [IN]	<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

#### 9.20.4.24 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> [IN]	<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> [IN]	<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> [IN]	<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> [IN]	<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> [IN]	<ul style="list-style-type: none"> <li>Maximum time to work on each fix (in seconds, max 255)</li> </ul>
<i>fixCount</i> [IN]	<ul style="list-style-type: none"> <li>Count of position fix requests for this session (must be at least 1)</li> </ul>
<i>fixInterval</i> [IN]	<ul style="list-style-type: none"> <li>interval between position fix requests (in seconds)</li> </ul>
<i>fixAccuracy</i> [IN]	<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

**9.20.4.25 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

## 9.21 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)

#### 9.21.1 Detailed Description

Quality of Service API function prototypes.

## 9.21.2 Macro Definition Documentation

9.21.2.1 `#define MAX_QOS_FILTER_TLV 25`

9.21.2.2 `#define MAX_QOS_SPEC_PER_APN (10)`

## 9.21.3 Function Documentation

9.21.3.1 `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[IN]</i>	Qos identifier Index identifying the QoS flow that has been negotiated
out	<i>pStatus[OUT]</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

9.21.3.2 `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[IN]</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[OUT]</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

### 9.21.3.3 ULONG SLQSQosGetNetworkStatus ( BYTE *instance*, BYTE \* *pStatus* )

Queries whether the device is currently on a network that supports QoS

## Parameters

	<i>instance</i> [IN]	<ul style="list-style-type: none"> <li>QMI instance</li> </ul>
out	<i>pStatus</i> [OUT]	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

### 9.21.3.4 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>	<i>pSz</i> 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

### 9.21.3.5 ULONG SLQSQosModify ( BYTE *instance*, swiQosModifyReq \* *pReq* )

Resume one or more existing QoS flows

## Warning

NOT IMPLEMENTED

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pReq[IN]</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 values9.21.3.6 ULONG SLQSQosRel ( BYTE *instance*, swiQosIds \* *pQosIds* )

Release one or more existing QoS flows

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pQosIds[IN]</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 values9.21.3.7 ULONG SLQSQosReq ( BYTE *instance*, swiQosReq \* *pQosReq*, swiQosIds \* *pQosResp* )

Triggers QoS negotiation by providing QoS parameters

## Parameters

<i>instance[IN]</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
<i>pQoSReq[IN]</i>	<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

#### 9.21.3.8 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

#### 9.21.3.9 ULONG SLQSQosResume ( BYTE *instance*, *swiQosIds* \* *pQosIds* )

Resume one or more existing QoS flows

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pQoslds[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQoslds</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

#### 9.21.3.10 ULONG SLQSQoSSuspend ( BYTE *instance*, swiQoslds \* *pQoslds* )

Suspend one or more existing QoS flows

## Parameters

in	<i>instance</i>	<ul style="list-style-type: none"> <li>• QMI instance</li> </ul>
	<i>pQoslds[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">swiQoslds</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

#### 9.21.3.11 ULONG SLQSQoSSwiReadApnExtraParams ( BYTE *instance*, ULONG *apnId*, sApnExtraParams \* *pApnExtraParams* )

Queries extra APN parameters that are not reported by existing QCT QMI service

## 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>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

### 9.21.3.12 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	instance	<ul style="list-style-type: none"> <li>QMI instance</li> </ul>
in	apnId	<ul style="list-style-type: none"> <li>APN id</li> </ul>
out	pQosStat	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

## 9.22 qaGobiApiRms.h File Reference

Remote Management Service API function prototypes.

## Functions

- [ULONG GetSMSWake](#) ( [ULONG](#) \*pEnabled, [ULONG](#) \*pWakeMask )
- [ULONG SetSMSWake](#) ( [ULONG](#) bEnable, [ULONG](#) wakeMask )

### 9.22.1 Detailed Description

Remote Management Service API function prototypes.

### 9.22.2 Function Documentation

#### 9.22.2.1 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

### 9.22.2.2 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

## 9.23 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)`

### 9.23.1 Detailed Description

Specific Absorption Rate API function prototypes.

### 9.23.2 Enumeration Type Documentation

#### 9.23.2.1 enum `eQMISARRFState`

This enum contains the SAR RF States

#### Parameters

SAR	RF State
	<ul style="list-style-type: none"> <li><code>QMI_SAR_RF_STATE_DEFAULT = 0</code></li> <li><code>QMI_SAR_RF_STATE_1</code></li> <li><code>QMI_SAR_RF_STATE_2</code></li> <li><code>QMI_SAR_RF_STATE_3</code></li> <li><code>QMI_SAR_RF_STATE_4</code></li> <li><code>QMI_SAR_RF_STATE_5</code></li> <li><code>QMI_SAR_RF_STATE_6</code></li> <li><code>QMI_SAR_RF_STATE_7</code></li> <li><code>QMI_SAR_RF_STATE_8</code></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`**

### 9.23.3 Function Documentation

#### 9.23.3.1 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

#### 9.23.3.2 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

## 9.24 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 [slqssendasyncsmsparams\\_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 SLQSDeleteSMS](#) (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, BYTE \*pSmsOnIms)
- [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 SLQSModifySMSStatus](#) (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 SLQSSwiGetSMSStorage](#) ([ULONG](#) \*pSmsStorage)
- [ULONG SLQSSendLongSMS](#) ([ULONG](#) messageFormat, [ULONG](#) messageSize, [CHAR](#) \*pMessage, [BYTE](#) encodingScheme, [ULONG](#) \*pMessageFailureCode, [CHAR](#) \*pMobileNum, [BYTE](#) \*pSmsOnIMS)

### 9.24.1 Detailed Description

Short Message Service API function prototypes.

### 9.24.2 Macro Definition Documentation

9.24.2.1 `#define ABSOLUTE_VALIDITY 0x08`

9.24.2.2 `#define CONFIG_LEN 0x05`

9.24.2.3 `#define MAX_SMS_ROUTES 0x0A`

9.24.2.4 `#define NUM_OF_SET 0xFF`

9.24.2.5 `#define TIME_DATE_BUF 0x09`

9.24.2.6 `#define TIME_STAMP_BUF 0x08`

### 9.24.3 Typedef Documentation

9.24.3.1 `typedef struct _getIndicationRegResp getIndicationRegResp`

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 BYTEparameter 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>
-----------------------------	---

#### 9.24.3.2 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>

#### 9.24.3.3 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>
-------------------	---

## 9.24.3.4 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>

## 9.24.3.5 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>

## 9.24.3.6 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>

## 9.24.3.7 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>

## 9.24.4 Function Documentation

## 9.24.4.1 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[OUT]</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[OUT]</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

#### 9.24.4.2 ULONG SaveSMS ( ULONG *storageType*, ULONG *messageFormat*, ULONG *messageSize*, BYTE \* *pMessage*, ULONG \* *pMessageIndex* )

Saves an SMS message to device memory

## Parameters

<i>storageType[IN]</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>message-Format[IN]</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

#### 9.24.4.3 ULONG SendSMS ( ULONG *messageFormat*, ULONG *messageSize*, BYTE \* *pMessage*, ULONG \* *pMessageFailureCode*, BYTE \* *pSmsOnIms* )

Sends an SMS message for immediate over-the-air transmission

**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>• The length of the message contents in bytes</li> </ul>
<i>pMessage</i> [IN]	<ul style="list-style-type: none"> <li>• The message contents in PDU format contains SMS header and payload message</li> </ul>
<i>pSmsOnIms</i> [IN]	<ul style="list-style-type: none"> <li>• (Optional) SMS on IMS</li> <li>• The message is to be sent on IMS. <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>pMessageFailureCode</i> [OUT]	<ul style="list-style-type: none"> <li>• (Optional) Message Failure Code</li> <li>• pointer to message failure code. If cause code is not provided, then value will be 0xFFFFFFFF</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

**9.24.4.4    ULONG SetSMSCAddress ( CHAR \* *pSMSCAddress*, CHAR \* *pSMSCType* )**

Sets the SMS center address.

**Parameters**

<i>pSMSC-Address</i> [IN]	<ul style="list-style-type: none"> <li>The SMS center address represented as a NULL terminated string</li> </ul>
<i>pSMSCType</i> [IN]	<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

**9.24.4.5    ULONG SLQSCDMADecodeMTTextMsg ( struct cdmaMsgDecodingParams \* *pCdmaMsgDecodingParams* )**

Decodes text message to CDMA PDU message

**Parameters**

<i>pMsgToBe-EncodedCDMA</i> [IN/OUT]	<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

**9.24.4.6    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

#### 9.24.4.7 ULONG SLQSDelSms ( 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[IN]</i>	<ul style="list-style-type: none"> <li>• (Optional) message index</li> </ul>
<i>pMessageTag[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>
	Generated on Wed Aug 24 2016 10:19:30 for LinuxQMI SDK by Doxygen

<i>pMessage-Mode[IN]</i>	<ul style="list-style-type: none"> <li>• (Optional) message mode</li> <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

**9.24.4.8 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

**9.24.4.9 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

#### 9.24.4.10 **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</i> [ <i>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</i> [ <i>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</i> [ <i>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</i> [ <i>OUT</i> ]	<ul style="list-style-type: none"> <li>The message contents array</li> </ul>
<i>pMessageMode</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>(Optional) Message 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>

**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

#### 9.24.4.11 **ULONG SLQSGetSmsBroadcastConfig ( BYTE mode, qaQmi3GPPBroadcastCfgInfo \* pBroadcastConfig, qaQmi3GPP2BroadcastCfgInfo \* pCDMABroadcastConfig )**

Provides Information about the SMS BroadcastConfiguration

**Parameters**

<i>mode</i> [IN]	<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</i> [OUT]	<ul style="list-style-type: none"> <li>The data for 3GPP Broadcast Information(Optional).</li> </ul>
<i>pCDMA-Broadcast-Config</i> [OUT]	<ul style="list-style-type: none"> <li>The data for 3GPP2 Broadcast Information(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

#### 9.24.4.12 **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> [IN]	<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</i> [IN]	<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>• (Optional) Message Mode</li> <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

#### 9.24.4.13 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

#### 9.24.4.14 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

#### 9.24.4.15 ULONG SLQSM ModifySMSStatus ( 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> [IN]	<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> [I-N]	<ul style="list-style-type: none"> <li>• Message index</li> </ul>
<i>messageTag</i> [IN]	<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>• (Optional) Message Mode</li> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>
Generated on Wed Aug 24 2016 10:19:30 for LinuxQMISDK by Doxygen	

**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

**9.24.4.16 ULONG SLQSSendAsyncSMS ( slqssendasyncsmsparams\_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="#">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

**9.24.4.17 ULONG SLQSSendLongSMS ( ULONG messageFormat, ULONG messageSize, CHAR \* pMessage, BYTE encodingScheme, ULONG \* pMessageFailureCode, CHAR \* pMobileNum, BYTE \* pSmsOnIMS )**

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>message-Format[IN]</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[IN]</i>	<ul style="list-style-type: none"> <li>Message size of the input message text</li> </ul>

<i>pMessage[IN]</i>	<ul style="list-style-type: none"> <li>• Original message text</li> </ul>
<i>encoding-Scheme[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>
<i>pMessage-FailureCode[OUT]</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>pMobileNum[IN]</i>	<ul style="list-style-type: none"> <li>• Mobile number of the receiver</li> </ul>
<i>pSmsOnIMS[IN]</i>	<ul style="list-style-type: none"> <li>• A flag indicates whether SMS was sent through IMS</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, but 3GPP2 does not support multiple sms  
Timeout: 5 minutes

**9.24.4.18 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

**9.24.4.19 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

**9.24.4.20 ULONG SLQSSetSmsBroadcastActivation ( BYTE mode, BYTE broadcastActivate )**

Enables or disables the reception of broadcast SMS messages.

**Parameters**

<i>Mode</i> [IN]	<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> [IN]	<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

#### 9.24.4.21 **ULONG SLQSSetSmsBroadcastConfig** ( **BYTE** *mode*, **qaQmi3GPPBroadcastCfgInfo** \* *pBroadcastConfig*, **qaQmi3GPP2BroadcastCfgInfo** \* *pCDMABroadcastConfig* )

Sets the information about the SMS BroadcastConfiguration

## Parameters

<i>mode</i> [IN]	<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</i> [IN]	<ul style="list-style-type: none"> <li>• The data for 3GPP Broadcast Information(Optional).</li> </ul>
<i>pCDMA-Broadcast-Config</i> [IN]	<ul style="list-style-type: none"> <li>• The data for 3GPP2 Broadcast Information(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

#### 9.24.4.22 **ULONG SLQSSetSmsStorage** ( **BYTE** *smsStorage* )

Sets the SMS Storage on the device

## Parameters

<i>smsStorage</i> [IN]	<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

#### 9.24.4.23 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</i> [IN]	<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</i> [OUT]	<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

#### 9.24.4.24 ULONG SLQSSmsGetMessageProtocol ( smsMsgprotocolResp \* pMessageProtocol )

This API queries the message protocol currently in use for the WMS client.

## Parameters

<i>pMessage-Protocol</i>	[OUT] <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

#### 9.24.4.25 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

#### 9.24.4.26 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> [O-UT]	<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

**9.24.4.27** **ULONG** SLQSWCDMADecodeLongTextMsg ( struct wcdmaLongMsgDecodingParams \*  
*pWcdmaLongMsgDecodingParams* )

Decodes WCDMA Long SMS PDU message, returns structure filled with decoded parameters

**Parameters**

<i>pWcdmaMsg-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

**9.24.4.28** **ULONG** SLQSWCDMADecodeMTTextMsg ( struct wcdmaMsgDecodingParams \*  
*pWcdmaMsgDecodingParams* )

Decodes WCDMA PDU message, returns structure filled with decoded parameters

**Parameters**

<i>pWcdmaMsg-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

9.24.4.29 **ULONG** SLQSWCDMAEncodeMOTextMsg ( struct wcdmaMsgEncodingParams \*  
pWcdmaMsgEncodingParams )

Returns the encoded WCDMA PDU message.

#### Parameters

<i>pwcdmaMsg- 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

## 9.25 qaGobiApiSwi.h File Reference

SWI API function prototypes.

### Functions

- [ULONG](#) SLQSGetSdkVersion ( [CHAR](#) \*\*sdkversionpp )
- [ULONG](#) SLQSSendRawQMI ( [BYTE](#) \*pReqBuf, [USHORT](#) service, [USHORT](#) length, [ULONG](#) timeout, [BYTE](#) \*\*ppInParm, [USHORT](#) \*pParamLength )
- int [SLQSGetPidof](#) ( [CHAR](#) \*pProcName )

### 9.25.1 Detailed Description

SWI API function prototypes.

### 9.25.2 Function Documentation

9.25.2.1 int SLQSGetPidof ( [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

**9.25.2.2 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

**9.25.2.3 ULONG SLQSSendRawQMI ( BYTE \* pReqBuf, USHORT service, USHORT length, ULONG timeout, BYTE \*\* ppInParm, USHORT \* pParamLength )****9.26 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)

### 9.26.1 Detailed Description

M2M Audio Service API function prototypes.

### 9.26.2 Macro Definition Documentation

#### 9.26.2.1 `#define MAX_LEN_IFACE_TABLE 255`

### 9.26.3 Function Documentation

#### 9.26.3.1 [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

### 9.26.3.2 **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

### 9.26.3.3 **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

#### 9.26.3.4 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

#### 9.26.3.5 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

#### 9.26.3.6 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

**9.26.3.7 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

**9.26.3.8 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

**9.26.3.9 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

### 9.26.3.10 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

### 9.26.3.11 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

**9.27 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 SLQSOMADMSelectSelection](#) (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 SLQSOMADMSelectSelection2](#) (ULONG selection, ULONG \*pDeferTime, ULONG \*pRejectReason)
- [ULONG SLQSOMADMSetSettings3](#) ([SLQSOMADMSettingsReqParams3](#) \*pSLQSOMADMSettingsReqParams3)

**9.27.1 Detailed Description**

SWI Open Mobile Alliance Device Management Service API function prototypes SWI OMA-DM QMI Service revision 1.6.

## 9.27.2 Typedef Documentation

### 9.27.2.1 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(optional) <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(optional) <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(optional) <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.(optional)</li> </ul>
<i>pSource</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Vendor Name in ASCII(optional)</li> </ul>
<i>pPkgName-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.(optional)</li> </ul>
<i>pPkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in ASCII(optional)</li> </ul>
<i>pPkgDesc-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.(optional)</li> </ul>
<i>pPkgDescription</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Description in ASCII(optional)</li> </ul>
<i>pDateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.(optional)</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.(optional)</li> </ul>

<i>pTime</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Time String in ASCII(optional)</li> </ul>
<i>pSessionType</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter reflects the last session started for Sprint(optional) <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(optional) <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(optional) <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.

**9.27.2.2 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>• 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>
	Generated on Wed Aug 24 2016 10:19:30 for LinuxQMISDK by Doxygen

<i>pFOTA- Adownload[OUT]</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 - 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>• 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>• 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

**9.27.2.3 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

#### 9.27.2.4 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

**9.27.3 Function Documentation****9.27.3.1 ULONG SLQSOMADMCancelSession ( ULONG session )**

Cancels an ongoing OMA-DM session.

**Parameters**

<i>session</i> [IN]	<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

**9.27.3.2 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

### 9.27.3.3 ULONG SLQSOMADMGetSettings ( ULONG \* *pbOMADMEnabled*, ULONG \* *pbFOTAdownload*, ULONG \* *pbFOTAUpdate* )

Returns the OMA-DM settings.

## Parameters

<i>pbOMADM-Enabled</i> [OUT]	<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</i> [OUT]	<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 autodownload 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</i> [OUT]	<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

### 9.27.3.4 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

**9.27.3.5 ULONG SLQSOMADMSelectSelection ( 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

**9.27.3.6 ULONG SLQSOMADMSelectSelection2 ( 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

### 9.27.3.7 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

### 9.27.3.8 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

### 9.27.3.9 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

### 9.27.3.10 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

### 9.27.3.11 `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

## 9.28 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

### 9.28.1 Detailed Description

Network Access Service API Band Classes table.

### 9.28.2 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

#### 9.28.2.1 LTE Bands

- 28 - LTE Band Class 28
- 39 - LTE Band Class 39
- 40 - LTE Band Class 40
- 41 - LTE Band Class 41
- 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: 1850-1919.5; 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

## 9.29 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

### 9.29.1 Detailed Description

Call Control Return Reasons table.

### 9.29.2 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\_OUTGOINGINTEXTOTHOME - 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

## 9.30 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

### 9.30.1 Detailed Description

Wireless Data Service Call End Reasons.

### 9.30.2 Call end reason codes (Code - Reason)

#### 9.30.2.1 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

#### 9.30.2.2 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

#### 9.30.2.3 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 SNDCCP(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

#### 9.30.2.4 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

#### 9.30.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

#### 9.30.2.6 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

#### 9.30.2.7 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

#### 9.30.2.8 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 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE TXN - Light Radio Resource Controller Uplink data confirmation failure
- 1047 - LRRRC(LTE Radio Resource Control) UL DATA CNF FAILURE HO
- 1048 - LRRRC(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

#### 9.30.2.9 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 SND CP 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-3GPP 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

#### 9.30.2.10 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

#### 9.30.2.11 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

#### 9.30.2.12 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

## 9.31 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

### 9.31.1 Detailed Description

Carrier Codes table.

#### 9.31.2 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

## 9.32 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

### Macros

- `#define __GOBI_API_CODING_SCHEME_H__`

### 9.32.1 Detailed Description

Data Coding Scheme.

### 9.32.2 Call Control Result Reasons (Value - Name - Description)

#### 9.32.2.1 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

### 9.32.3 Coding Group Bits 7..4(0001)

#### 9.32.3.1 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.

### 9.32.4 Coding Group Bits 7..4(0010)

#### 9.32.4.1 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

### 9.32.5 Coding Group Bits 7..4(0011)

#### 9.32.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

### 9.32.6 Coding Group Bits 7..4(01xx)

#### 9.32.6.1 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

### 9.32.7 Coding Group Bits 7..4(1001)

#### 9.32.7.1 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

### 9.32.8 Coding Group Bits 7..4(1010..1101)

#### 9.32.8.1 Reserved coding groups

### 9.32.9 Coding Group Bits 7..4(1110)

#### 9.32.9.1 Defined by the WAP Forum

### 9.32.10 Coding Group Bits 7..4 (1111)

#### 9.32.10.1 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

#### 9.32.11 Macro Definition Documentation

9.32.11.1 `#define __GOBI_API_CODING_SCHEME_H__`

### 9.33 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

#### 9.33.1 Detailed Description

Position Determination Service API GPS Capability Codes.

#### 9.33.2 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

### 9.34 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

#### 9.34.1 Detailed Description

Device Management Service API Power Modes table.

### 9.34.2 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

Valid transitions for Power Modes

- Online to Low Power, Persistent low power, Factory test, Offline or Shut Down
- Low power to online, Persistent low power, Offline, or Shut Down
- Persistent low power to Online, Low power, Offline or Shut down
- Factory test to online
- Offline to Reset

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.35 qaGobiApiTableRadiolInterfaces.h File Reference

Network Access Service API Radio Interfaces table.

### 9.35.1 Detailed Description

Network Access Service API Radio Interfaces table.

### 9.35.2 Radio interface

#### 9.35.2.1 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

## 9.36 qaGobiApiTableRegionCodes.h File Reference

Region Codes table.

### 9.36.1 Detailed Description

Region Codes table.

### 9.36.2 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

## 9.37 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

### 9.37.1 Detailed Description

Voice Service Options.

### 9.37.2 Service Option codes (Code - Reason)

#### 9.37.2.1 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

## 9.38 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

### 9.38.1 Detailed Description

Voice Supplementary Service Information Classes.

### 9.38.2 Supplementary Service Information Classes (Value - Service Class)

- 0X00 - CLASS\_NONE
- 0X01 - CLASS\_VOICE
- 0X02 - CLASS\_DATA
- 0X04 - CLASS\_FAX
- 0X08 - CLASS\_SMS
- 0X10 - CLASS\_DATAACIRCUITSYNC
- 0X20 - CLASS\_DATAACIRCUITASYNC
- 0X40 - CLASS\_PACKETACCESS
- 0X80 - CLASS\_PADACCESS

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.39 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

### 9.39.1 Detailed Description

Swi Audio related tables.

### 9.39.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

### 9.39.3 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

## 9.40 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

### 9.40.1 Detailed Description

Update Complete Status table.

### 9.40.2 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

## 9.41 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

### 9.41.1 Detailed Description

Voice Service Call and supplementary services end reasons.

### 9.41.2 Voice Call and supplementary services end reason codes (Code - Reason)

#### 9.41.2.1 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

## 9.41.2.2 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

#### 9.41.2.3 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

#### 9.41.2.4 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

#### 9.41.2.5 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

#### 9.41.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

#### 9.41.2.7 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

#### 9.41.2.8 reject causes

- 237 - Service option is not supported

#### 9.41.2.9 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

## 9.42 qaGobiApiTmd.h File Reference

Thermal Mitigation Device API function prototypes.

### Data Structures

- struct [mitigationDevList](#)
- struct [TmdGetMitigationDevListResp](#)
- struct [TmdGetMitigationLvlReq](#)
- struct [TmdGetMitigationLvlResp](#)
- struct [TmdRegNotMitigationLvlReq](#)
- struct [TmdDeRegNotMitigationLvlReq](#)
- struct [TmdMitigationLvlIndReq](#)

## Macros

- `#define MAX_MITIGATION_DEV_LIST_LEN 255`
- `#define MAX_MITIGATION_DEV_ID_LEN 255`

## Functions

- `ULONG SLQSTmdGetMitigationDevList (TmdGetMitigationDevListResp *pTmdGetMitigationDevListResp)`
- `ULONG SLQSTmdGetMitigationLvl (TmdGetMitigationLvlReq *pTmdGetMitigationLvlReq, TmdGetMitigationLvlResp *pTmdGetMitigationLvlResp)`
- `ULONG SLQSTmdRegNotMitigationLvl (TmdRegNotMitigationLvlReq *pTmdRegNotMitigationLvlReq)`
- `ULONG SLQSTmdDeRegNotMitigationLvl (TmdDeRegNotMitigationLvlReq *pTmdDeRegNotMitigationLvlReq)`

### 9.42.1 Detailed Description

Thermal Mitigation Device API function prototypes.

### 9.42.2 Macro Definition Documentation

9.42.2.1 `#define MAX_MITIGATION_DEV_ID_LEN 255`

9.42.2.2 `#define MAX_MITIGATION_DEV_LIST_LEN 255`

### 9.42.3 Function Documentation

9.42.3.1 `ULONG SLQSTmdDeRegNotMitigationLvl ( TmdDeRegNotMitigationLvlReq * pTmdDeRegNotMitigationLvlReq )`

Deregisters the notification for mitigation device level changes.

#### Parameters

<i>pTmdDeRegNotMitigationLvlReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">TmdDeRegNotMitigationLvlReq</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

9.42.3.2 `ULONG SLQSTmdGetMitigationDevList ( TmdGetMitigationDevListResp * pTmdGetMitigationDevListResp )`

Used by the control point to gets the list of mitigation devices from the remote endpoint.

#### Parameters

<i>pTmdGetMitigationDevListResp[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">TmdGetMitigationDevListResp</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

#### 9.42.3.3 ULONG SLQSTmdGetMitigationLvl ( TmdGetMitigationLvlReq \* pTmdGetMitigationLvlReq, TmdGetMitigationLvlResp \* pTmdGetMitigationLvlResp )

Gets the thermal mitigation level for the mitigation device

## Parameters

<i>pTmdGetMitigationLvlReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">TmdGetMitigationLvlReq</a> for more information</li> </ul>
<i>pTmdGetMitigationLvlResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">TmdGetMitigationLvlResp</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

#### 9.42.3.4 ULONG SLQSTmdRegNotMitigationLvl ( TmdRegNotMitigationLvlReq \* pTmdRegNotMitigationLvlReq )

Registers for notification of mitigation device level changes.

## Parameters

<i>pTmdRegNotMitigationLvlReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">TmdRegNotMitigationLvlReq</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

## 9.43 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 [unblockUIMPIN](#)
- struct [UIMUnblockPinReq](#)
- 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](#)
- struct [UIMSlotStatus](#)
- struct [UIMSlotsStatus](#)
- struct [UIMGetSlotsStatusResp](#)
- struct [UIMSwitchSlotReq](#)
- struct [personalizationStatus](#)
- struct [UIMGetConfigurationReq](#)
- struct [UIMGetConfigurationResp](#)

## 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
- #define [MAX\\_ICCID\\_LENGTH](#) 255
- #define [MAX\\_SLOTS\\_STATUS](#) 255
- #define [MAX\\_ACTIVE\\_PERS\\_FEATURES](#) 12

## 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 SLQSUIUnblockPin](#) (UIMUnblockPinReq \*pUIMUnblockPinReq, [UIMPinResp](#) \*pUIMUnblockPinResp)
- [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)
- [ULONG SLQSUIGetSlotsStatus](#) (UIMGetSlotsStatusResp \*pResp)
- [ULONG SLQSUISwitchSlot](#) (UIMSwitchSlotReq \*pReq)
- [ULONG SLQSUIGetConfiguration](#) (UIMGetConfigurationReq \*pUIMGetConfigurationReq, [UIMGetConfigurationResp](#) \*pUIMGetConfigurationResp)

### 9.43.1 Detailed Description

Uim Service API function prototypes.

### 9.43.2 Macro Definition Documentation

9.43.2.1 `#define MAX_ACTIVE_PERS_FEATURES 12`

9.43.2.2 `#define MAX_CONTENT_LENGTH 1024`

9.43.2.3 `#define MAX_DESCRIPTION_LENGTH 255`

9.43.2.4 `#define MAX_ICCID_LENGTH 255`

9.43.2.5 `#define MAX_NO_OF_APPLICATIONS 10`

9.43.2.6 `#define MAX_NO_OF_SLOTS 5`

9.43.2.7 `#define MAX_PATH_LENGTH 255`

9.43.2.8 `#define MAX_PUK_LENGTH 8`

9.43.2.9 `#define MAX_SLOTS_STATUS 255`

### 9.43.3 Function Documentation

9.43.3.1 **ULONG SLQSUIMAuthenticate ( UIMAuthenticateReq \* *pUIMAuthenticateReq*, UIMAuthenticateResp \* *pUIMAuthenticateResp* )**

This API executes the authentication algorithm on the card.

#### Parameters

<i>pUIM-Authenticate-Req</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMAuthenticateReq</a> for more information.</li> </ul>
<i>pUIM-Authenticate-Resp</i> [OUT]	<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

9.43.3.2 **ULONG SLQSUIMChangePin ( UIMChangePinReq \* *pUIMChangePinReq*, UIMPinResp \* *pUIMChangePinResp* )**

This API changes the value of the specified PIN.

## Parameters

<i>pUIMChange-PinReq[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMChangePinReq</a> for more information.</li> </ul>
<i>pUIMChange-PinResp[OUT]</i>	<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.

### 9.43.3.3 ULONG SLQSUIDepersonalization ( UIMDepersonalizationReq \* pUIMDepersonalizationReq, UIMDepersonalizationResp \* pUIMDepersonalizationResp )

This API de-activates or unblocks the personalization on the phone.

## Parameters

<i>pUIM-Depersonalization-Req[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMDepersonalizationReq</a> for more information.</li> </ul>
<i>pUIM-Depersonalization-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.

#### 9.43.3.4 ULONG SLQSUIMEventRegister ( UIIMEventRegisterReqResp \* pUIIMEventRegisterReqResp )

This API Registers for event notifications from the card.

##### Parameters

<i>pUIIMEventRegisterReqResp</i> [IN/OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIIMEventRegisterReqResp</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.

#### 9.43.3.5 ULONG SLQSUIMGetCardStatus ( UIIMGetCardStatusResp \* pUIIMGetCardStatusResp )

This API retrieves the current status of the card.

##### Parameters

<i>pUIIMGetCardStatusResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIIMGetCardStatusResp</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(activated) 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.

Please use \ref SLQSUIMGetSlotsStatus to retrieves active and inactivate SIMs status.

#### 9.43.3.6 **ULONG** SLQSUIMGetConfiguration ( **UIMGetConfigurationReq** \* *pUIMGetConfigurationReq*, **UIMGetConfigurationResp** \* *pUIMGetConfigurationResp* )

This API Gets the modem configuration for the UIM module.

##### Parameters

<i>pUIMGetConfigurationReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetConfigurationReq</a> for more information.</li> </ul>
<i>pUIMGetConfigurationResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetConfigurationResp</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

#### 9.43.3.7 **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</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetFileAttributesReq</a> for more information.</li> </ul>
<i>pUIMGetFileAttributesResp</i> [OUT]	<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.

#### 9.43.3.8 ULONG SLQSUIMGetSlotsStatus ( UIMGetSlotsStatusResp \* *pResp* )

This API Retrieves the current of the physical and logical slots.

##### Parameters

<i>pResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">UIMGetSlotsStatusResp</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

#### 9.43.3.9 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.

#### 9.43.3.10 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.

#### 9.43.3.11 ULONG SLQSUIReadTransparent ( UIMReadTransparentReq \* *pUIMReadTransparentReq*, UIMReadTransparentResp \* *pUIMReadTransparentResp* )

This API executes the Read Transparent algorithm on the card.

## Parameters

<i>pUIMReadTransparentReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMReadTransparentReq</a> for more information.</li> </ul>
<i>pUIMReadTransparentResp</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.

#### 9.43.3.12 ULONG SLQSUIRefreshComplete ( UIMRefreshCompleteReq \* *pUIMRefreshCompleteReq* )

This API invoked when the client has finished the Refresh procedure.

## Parameters

<i>pUIMRefresh-CompleteReq[IN]</i>	<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

#### 9.43.3.13 ULONG SLQSUIMRefreshGetLastEvent ( UIMRefreshGetLastEventReq \* pUIMRefreshGetLastEventReq, UIMRefreshGetLastEventResp \* pUIMRefreshGetLastEventResp )

This API provides the ability to retrieve the last refresh event.

## Parameters

<i>pUIMRefresh-GetLastEvent-Req[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">UIMRefreshGetLastEventReq</a> for more information.</li> </ul>
<i>pUIMRefresh-GetLastEvent-Resp[OUT]</i>	<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 )

#### 9.43.3.14 ULONG SLQSUIRefreshOK ( UIMRefreshOKReq \* pUIMRefreshOKReq )

This API Enables the client to indicate whether it is OK to start the Refresh procedure.

##### Parameters

<i>pUIMRefreshOKReq</i> [IN]	<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.

#### 9.43.3.15 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 SLQSUIRefreshSetRefreshCallBack for the events to be registered.

### 9.43.3.16 ULONG SLQSUIReset ( )

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 SLQSUIEventReg API to start receiving the events again. This would mean that the callback registrations would be reset after this API.

### 9.43.3.17 ULONG SLQSUISetPinProtection ( UIMSetPinProtectionReq \* *pUIMSetPinProtectionReq*, UIMPinResp \* *pUIMSetPinProtectionResp* )

This API enables or disables the protection of the UIM contents by a specific PIN.

#### Parameters

<i>pUIMSetPinProtectionReq</i> [I-N]	<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 SLQSUISetPinProtectionCallback indication.

**9.43.3.18    ULONG SLQSUIMSwitchSlot ( UIMSwitchSlotReq \* *pReq* )**

This API Switches the binding between a logical slot and a physical slot.

**Parameters**

<i>pReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">UIMSwitchSlotReq</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

Please wait at least one second before this API call again.

**9.43.3.19    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 SLQSUIMUnblockPinCallback.

### 9.43.3.20 ULONG SLQSUIVerifyPin ( 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 SLQSUIVerifyPinCallback.

## 9.44 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)

### 9.44.1 Detailed Description

Voice Service API function prototypes.

### 9.44.2 Macro Definition Documentation

9.44.2.1 `#define MAX_CALL_NO_LEN 81`

9.44.2.2 `#define MAX_DESCRIPTION_LENGTH 255`

9.44.2.3 `#define MAX_NO_OF_CALLS 20`

9.44.2.4 `#define MAXUSSDLENGTH 182`

9.44.2.5 `#define PASSWORD_LENGTH 4`

### 9.44.3 Enumeration Type Documentation

9.44.3.1 `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**

#### 9.44.4 Function Documentation

##### 9.44.4.1 **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

##### 9.44.4.2 **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

9.44.4.3 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

9.44.4.4 ULONG SLQSOriginateUSSD ( 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

9.44.4.5 ULONG SLQSVoiceALSSelectLine ( voiceALSSelectLineInfo \* *pVoiceALSSelectLineInfo* )

This API allows the user to select the preferred line.

## Parameters

<i>pVoiceALS-SelectLineInfo</i> [IN]	<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.

#### 9.44.4.6 ULONG SLQSVoiceALSSetLineSwitching ( voiceALSSetLineSwitchInfo \* pVoiceALSSetLineSwitchInfo )

This API sets the line switch setting on the card.

## Parameters

<i>pVoiceALSSetLineSwitchInfo</i> [IN]	<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.

#### 9.44.4.7 ULONG SLQSVoiceAnswerCall ( voiceAnswerCall \* pVoiceAnswerCall )

Answers an incoming voice call.

## Parameters

<i>pVoiceAnswerCall</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">voiceAnswerCall</a></li> </ul>
	<ul style="list-style-type: none"> <li>See <a href="#">voiceAnswerCall</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 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.

**9.44.4.8 ULONG SLQSVoiceBindSubscription ( voiceBindSubscriptionInfo \* pVoiceBindSubscriptionInfo )**

This API binds a subscription type to a specific voice client ID.

**Parameters**

<i>pVoiceBindSubscriptionInfo</i> [IN]	<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.

**9.44.4.9 ULONG SLQSVoiceBurstDTMF ( voiceBurstDTMFInfo \* pBurstDTMFInfo )**

Sends a burst Dual-Tone Multi frequency (DTMF) (applicable only for 3GPP2)

**Parameters**

<i>pBurstDTMFInfo</i> [IN/OUT]	<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.

#### 9.44.4.10 **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.

#### 9.44.4.11 **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.

#### 9.44.4.12 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.

#### 9.44.4.13 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.

#### 9.44.4.14 **ULONG SLQSVoiceGetCallForwardingStatus ( voiceGetCallFWReq \* pVoiceGetCallFWReq, voiceGetCallFWResp \* pVoiceGetCallFWResp )**

Queries the status of Call Forwarding Supplementary Service.

**Parameters**

<i>pVoiceGetCallFWReq[IN]</i>	<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[OUT]</i>	<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.

#### 9.44.4.15 **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>pGetCallInfo-Req[IN]</i>	<ul style="list-style-type: none"> <li>See <a href="#">voiceCallInfoReq</a> for more information.</li> </ul>
<i>pGetCallInfo-Resp[OUT]</i>	<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.

#### 9.44.4.16 ULONG SLQSVoiceGetCallWaiting ( voiceGetCallWaitInfo \* pVoiceGetCallWaitInfo )

Queries the status of Call Waiting Supplementary Service (applicable only for 3GPP).

## Parameters

<i>pVoiceGetCall-WaitInfo[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.

#### 9.44.4.17 ULONG SLQSVoiceGetCLIP ( voiceGetCLIPResp \* pVoiceGetCLIPResp )

Queries the status of the Calling Line Identification Presentation (CLIP) supplementary service (applicable only for 3GPP).

## Parameters

<i>pVoiceGetCLIP-Resp[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.

#### 9.44.4.18 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.

#### 9.44.4.19 ULONG SLQSVoiceGetCNAP ( voiceGetCNAPResp \* pVoiceGetCNAPResp )

Queries the status of the Calling Name Presentation(CNAP) supplementary service (applicable only for 3GPP).

##### Parameters

<i>pVoiceGetCNAPResp</i> [OUT]	<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.

#### 9.44.4.20 ULONG SLQSVoiceGetCOLP ( voiceGetCOLPResp \* pVoiceGetCOLPResp )

Queries the status of the Connected Line Identification Presentation (COLP) supplementary service (applicable only for 3GPP).

##### Parameters

<i>pVoiceGetCOLPResp</i> [OUT]	<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.

#### 9.44.4.21 **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.

#### 9.44.4.22 **ULONG SLQSVoiceGetConfig ( voiceGetConfigReq \* pVoiceGetConfigReq, voiceGetConfigResp \* pVoiceGetConfigResp )**

This API retrieves various configuration parameters that control the modem behavior related to circuit switched services.

##### Parameters

<i>pVoiceGetConfigReq</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>pVoiceGetConfigResp</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.

**9.44.4.23 ULONG SLQSVoiceIndicationRegister ( voiceIndicationRegisterInfo \* pVoiceIndicationRegisterInfo )**

Sets the registration state for different QMI\_VOICE indications for the requesting control point

**Parameters**

<i>pVoice-Indication-RegisterInfo[IN]</i>	<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.

**9.44.4.24 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[IN]</i>	<ul style="list-style-type: none"> <li>Request structure of to manage calls.</li> </ul>
<i>pVoiceManageCallsResp[OUT]</i>	<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"

**9.44.4.25 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>pVoiceOrigUSSDNoWaitInfo</i> [IN]	<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.

**9.44.4.26 ULONG SLQSVoiceSendFlash ( voiceFlashInfo \* pFlashInfo )**

This API sends a simple flash message. Applicable only for 3GPP2 devices.

**Parameters**

<i>pFlashInfo</i> [IN/OUT]	<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.

#### 9.44.4.27 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.

#### 9.44.4.28 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.

#### 9.44.4.29 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

#### 9.44.4.30 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>
	Generated on Wed Aug 24 2016 10:19:30 for LinuxQMISDK by Doxygen

## 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

#### 9.44.4.31 ULONG SLQSVoiceStartContDTMF ( voiceContDTMFInfo \* pContDTMFInfo )

Starts a continuous DTMF.

## Parameters

<i>pContDTMF-Info</i> [IN/OUT]	<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.

#### 9.44.4.32 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.

## 9.45 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](#)
- struct [WdsDHCPv4ProfileId](#)
- struct [WdsDHCPv4HWConfig](#)
- struct [WdsDHCPv4Option](#)
- struct [WdsDHCPv4OptionList](#)
- struct [WdsDHCPv4Config](#)
- struct [WdsClientLeaseChange](#)

## Macros

- [#define IPV6\\_ADDRESS\\_ARRAY\\_SIZE 8](#)

## Typedefs

- typedef struct [dataBearerTechnology](#) [QmiWSDDataBearerTechnology](#)
- typedef struct [dataBearers](#) [QmiWSDDataBearers](#)
- 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 (WdsIpAddrInfoReq *)`
- `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](#) \*pPDPType, [ULONG](#) \*pIPAddress, [ULONG](#) \*pPrimaryDNS, [ULONG](#) \*pSecondaryDNS, [ULONG](#) \*pAuthentication, [CHAR](#) \*pName, [CHAR](#) \*pAPNName, [CHAR](#) \*pUsername, [CHAR](#) \*pPassword)
- [ULONG SLQSGetProfile](#) ([ULONG](#) profileType, [BYTE](#) profileId, [ULONG](#) \*pPDPType, [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 SLQSGetDataBearerTechnologyExt](#) ([DataBearerTechExt](#) \*pDataBearerTech, [BYTE](#) instance)
- [ULONG SLQSGetCurrentChannelRate](#) ([WDSSWICurrentChannelRates](#) \*pRates, [BYTE](#) instance)
- [ULONG SLQSSetLoopback](#) ([WDSSetLoopbackData](#) \*pReq)
- [ULONG SLQSSGetLoopback](#) ([WDSGetLoopbackData](#) \*data)
- [ULONG GetDefaultProfileNum](#) ([BYTE](#) profile\_type, [BYTE](#) profile\_family, [BYTE](#) \*pProfile\_no)
- [ULONG SetDefaultProfileNum](#) ([BYTE](#) profile\_type, [BYTE](#) profile\_family, [BYTE](#) profile\_index)
- [ULONG SLQSSetDHCIPv4ClientConfig](#) ([WdsDHCIPv4Config](#) \*pReq)
- [ULONG SLQSSGetDHCIPv4ClientConfig](#) ([WdsDHCIPv4Config](#) \*pReqResp)

### 9.45.1 Detailed Description

Wireless Data Service API function prototypes.

## 9.45.2 Macro Definition Documentation

### 9.45.2.1 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

## 9.45.3 Typedef Documentation

### 9.45.3.1 `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>

### 9.45.3.2 `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 <code>eQCWWAN_ERR_QMI_EXTENDED_INTERNAL</code> 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>

### 9.45.3.3 `typedef union WdsProfileParam QmiProfileInfo`

This union [WdsProfileParam](#) consist of [Profile3GPP](#) and [Profile3GPP2](#) out of which one will be used to create profile.

### 9.45.3.4 `typedef struct dataBearers QmiWDSDataBearers`

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>pCurData-Bearer-Technology[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>pLastCallData-Bearer-Technology[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>

#### 9.45.3.5 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>

### 9.45.3.6 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 (EM/MC73xx or earlier)</li> <li>– Optional parameter with possible values 1-24 (EM/MC74xx onwards)</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> <li>– The new equivalent option for "pLTEAttachProfile" on 74xx modems is "pLTEAttachProfile-List". Please provide attach profiles in order of decreasing priority in this list.</li> </ul> </li> </ul>

<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>
---------------------------------	--

## 9.45.4 Enumeration Type Documentation

### 9.45.4.1 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***

## 9.45.5 Function Documentation

### 9.45.5.1 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

### 9.45.5.2 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

**9.45.5.3** **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

#### 9.45.5.4 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</i> [O-UT]	<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

#### 9.45.5.5 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[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>

## 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

**9.45.5.6** `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>pIPAddressv4</i> [-OUT]	<ul style="list-style-type: none"> <li>Preferred IPv4 addr to be assigned to device</li> </ul>
<i>pPrimaryDN-Sv4</i> [OUT]	<ul style="list-style-type: none"> <li>Primary DNS ipv4 address preference</li> </ul>
<i>pSecondaryDN-Sv4</i> [OUT]	<ul style="list-style-type: none"> <li>Secondary DNS ipv4 address preference</li> </ul>
<i>pIPAddressv6</i> [-OUT]	<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;] IPv6 address: 1234:2A01:.....:5678 U0 corresponds to 1234 U1 corresponds to 2A01 ----- ----- U7 corresponds to 5678</li> </ul>
<i>pPrimaryDN-Sv6</i> [OUT]	<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

#### 9.45.5.7 ULONG GetDefaultProfileNum ( BYTE *profile\_type*, BYTE *profile\_family*, BYTE \* *pProfile\_no* )

This API to Get default profile number

## Parameters

<i>profile_type</i>	[IN] <ul style="list-style-type: none"> <li>• 0 - 3GPP</li> <li>• 1 - 3GPP2</li> </ul>
<i>profile_family</i>	[IN] <ul style="list-style-type: none"> <li>• 0 - Embedded</li> <li>• 1 - Tethered</li> </ul>
<i>pProfile_no</i>	[OUT]

## 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

#### 9.45.5.8 ULONG GetDormancyState ( ULONG \* pDormancyState, BYTE instance )

Returns the dormancy state of the current packet data session when connected.

## Parameters

<i>pDormancy-State</i> [OUT]	<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

#### 9.45.5.9 ULONG GetIPAddressLTE ( WdsIpAddressInfoReq \* )

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.

**9.45.5.10 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

**9.45.5.11 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

**9.45.5.12** `ULONG GetMobileIPProfile ( BYTE index, BYTE * pEnabled, ULONG * pAddress, ULONG * pPrimaryHA, ULONG * pSecondaryHA, BYTE * pRevTunneling, BYTE naiSize, CHAR * pNAI, ULONG * pHASPI, ULONG * pAAASPI, ULONG * pHAState, 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>pHASState[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

**9.45.5.13** **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

**9.45.5.14** **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

**9.45.5.15    ULONG GetSessionDuration (    ULONGLONG \* *pDuration*, BYTE *instance* )**

Returns the duration of the current packet data session.

**Parameters**

<i>pDuration</i> [OUT]	<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

**9.45.5.16    ULONG GetSessionState (    ULONG \* *pState*, BYTE *instance* )**

Returns the state of the current packet data session.

**Parameters**

<i>pState</i> [OUT]	<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

- 9.45.5.17 **ULONG** iGetByteTotals ( **ULONG** \* *pv4sessionId*, **ULONG** \* *pv6sessionId*, struct WdsByteTotalsElmnts \* *pByteTotalsElmnt* )
- 9.45.5.18 **ULONG** iGetConnectionRate ( **ULONG** \* *pv4sessionId*, **ULONG** \* *pv6sessionId*, struct WdsConnectionRateElmnts \* *pConnectionRateElmnt* )
- 9.45.5.19 **ULONG** iGetPacketStatistics ( **ULONG** \* *pV4sessionId*, **ULONG** \* *pV6sessionId*, struct WdsPktStatisticsReq \* *pStatMask*, struct WdsPktStatisticsElmnts \* *pPktStatisticsElmnt* )
- 9.45.5.20 **ULONG** iSLQSMISetIPFamilyPreference ( **BYTE** *IPFamilyPreference*, **BYTE** *instance* )
- 9.45.5.21 **ULONG** RMSetTransferStatistics ( **swiRMTrasferStaticsReq** \* *pSwiRMTrasferStaticsReq* )

This API request the device to fetch current data system transfer Statistics.

#### Parameters

<i>pSwiRMTrasferStaticsReq</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">swiRMTrasferStaticsReq</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

- 9.45.5.22 **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

#### 9.45.5.23 ULONG SetAutoconnect ( ULONG *setting* )

Sets the auto connect data session setting.

##### Parameters

<i>setting</i> [IN]	<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

#### 9.45.5.24 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</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 addr 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.

**9.45.5.25** **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</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>pIPAddressv4</i> [1-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>pIPAddressv6</i> [1-N]	<ul style="list-style-type: none"> <li>• Preferred IPv6 address 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> [1-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

Deprecated, please use SetDefaultProfileLTEV2 instead  
 Technology Supported: LTE  
 Timeout: 2 seconds

**9.45.5.26** **ULONG** SetDefaultProfileLTEV2 ( **ULONG** *profileType*, **ULONG** \* *pPDPTType*, **ULONG** \* *pIPAddressv4*, **ULONG** \* *pPrimaryDNSv4*, **ULONG** \* *pSecondaryDNSv4*, **USHORT** \* *pIPAddressv6*, **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[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>pPrimaryDNSv4[IN]</i>	<ul style="list-style-type: none"> <li>Primary DNS ipv4 address preference (optional)</li> </ul>
<i>pSecondaryDNSv4[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 addr to be assigned to device (optional)</li> </ul>
<i>pPrimaryDNSv6[IN]</i>	<ul style="list-style-type: none"> <li>Primary DNS ipv6 address preference (optional)</li> </ul>
<i>pSecondaryDNSv6[IN]</i>	<ul style="list-style-type: none"> <li>Secondary DNS ipv6 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

Technology Supported: LTE  
Timeout: 2 seconds  
Replaces deprecated Function SetDefaultProfileLTE

#### 9.45.5.27 ULONG SetDefaultProfileNum ( BYTE profile\_type, BYTE profile\_family, BYTE profile\_index )

This API to Set default profile number

#### Parameters

<i>profile_type</i>	[IN] <ul style="list-style-type: none"> <li>• 0 - 3GPP</li> <li>• 1 - 3GPP2</li> </ul>
---------------------	--

<i>profile_family</i>	[IN] <ul style="list-style-type: none"> <li>• 0 - Embedded</li> <li>• 1 - Tethered</li> </ul>
<i>profile_index</i>	[IN]

**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

**9.45.5.28 ULONG SetMobileIP ( ULONG mode )**

Sets the current mobile IP setting.

**Parameters**

<i>mode</i> [IN]	<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

**9.45.5.29 ULONG SetMobileIPParameters ( CHAR \* pSPC, ULONG \* pMode, BYTE \* pRetryLimit, BYTE \* pRetryInterval, BYTE \* pReRegPeriod, BYTE \* pReRegTraffic, BYTE \* pHAAuthenticator, 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>pHA-Authenticator</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

**9.45.5.30** **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 )**

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>pMNAAA</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

9.45.5.31 ULONG SLQSAutoConnect ( struct slqsautoconnect \* *pacreq* )

Returns auto connect settings

## Parameters

<i>slqsautoconnect</i> [-IN]	<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

#### 9.45.5.32 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</i> [IN]	<ul style="list-style-type: none"> <li>• SLQS Create profile Information</li> </ul>
<i>pResp</i> [OUT]	<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

#### 9.45.5.33 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> [IN]	<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>pExtendedErrorCode</i> [OUT]	<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

#### 9.45.5.34 ULONG SLQSGet3GPPConfigItem ( slqs3GPPConfigItem \* pSLQS3GPPConfigItem )

Reads the 3gpp configuration item.

## Parameters

<i>pSLQS3GPPConfigItem</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

#### 9.45.5.35 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

#### 9.45.5.36 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>pConnectionRate</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

#### 9.45.5.37 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

**9.45.5.38    ULONG SLQSGGetCurrentChannelRate ( 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

**9.45.5.39    ULONG SLQSGetDataBearerTechnology ( QmiWSDDataBearers \* *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>qmiWSDDataBearers</i> [OUT]	<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

**9.45.5.40 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

**9.45.5.41 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

#### 9.45.5.42 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</i> [IN]	– See <a href="#">WdsPktStatisticsReq</a> for more information
<i>pPktStatistics</i> [OUT]	– 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

#### 9.45.5.43 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 (EM/MC73xx or earlier)</li> <li>Value between 1 - 24 (EM/MC74xx onwards)</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>pIPAddress</i> [OUT]	<ul style="list-style-type: none"> <li>Preferred IPv4 address to be assigned to device</li> </ul>
<i>pPrimaryDNS</i> [OUT]	<ul style="list-style-type: none"> <li>Primary DNS ipv4 address preference</li> </ul>
<i>pSecondaryDNS</i> [OUT]	<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 <code>eWDS_ERR_PROFILE_REG_XXX</code>.</li> <li>• Error code will only be present if error code <code>eQCWWAN_ERR_QMI_EXTENDED_INTERNAL</code> 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

**9.45.5.44    ULONG SLQSGetProfileSettings ( GetProfileSettingIn \* *pReq*, GetProfileSettingOut \* *pResp* )**

Retrieves a profile(3GPP/3GPP2) with the specified parameters.

**Parameters**

<i>pReq</i> [IN]	<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</i> [OUT]	<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

**9.45.5.45    ULONG SLQSGetRuntimeSettings ( struct WdsRunTimeSettings \* *pRunTimeSettings* )**

Returns the packet data session settings currently in use.

**Parameters**

<i>pRunTimeSettings</i> [OUT]	<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

**9.45.5.46    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

#### 9.45.5.47 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

**9.45.5.48 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

**9.45.5.49 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

**9.45.5.50 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 (EM/MC73xx or earlier)</li> <li>– Value between 1 - 24 (EM/MC74xx onwards)</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 address 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[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[IN]</i>	<ul style="list-style-type: none"> <li>• profile Name (optional)</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 (optional)</li> <li>• If value is NULL or omitted, then subscription default value will be requested.</li> </ul>
<i>pUsername[IN]</i>	<ul style="list-style-type: none"> <li>• Username used during network authentication (optional)</li> </ul>
<i>pPassword[IN]</i>	<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.

#### 9.45.5.51 ULONG SLQSSGetDHCPv4ClientConfig ( WdsDHCPv4Config \* *pReqResp* )

This API gets the DHCP Client V4 Configuration.

## Parameters

<i>pReq</i>	[IN] • See <a href="#">WdsDHCPv4Config</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: 2 seconds\n

#### 9.45.5.52 ULONG SLQSSGetLoopback ( WDSGetLoopbackData \* *data* )

This API to Get the value of loopback mode and multiplier.

## Parameters

<i>pReq</i>	[IN] • See <a href="#">WDSGetLoopbackData</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: 2 seconds\n

#### 9.45.5.53 ULONG SLQSSSetDHCPv4ClientConfig ( WdsDHCPv4Config \* *pReq* )

This API sets the DHCP Client V4 Configuration.

## Parameters

<i>pReq</i>	[IN] • See <a href="#">WdsDHCPv4Config</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: 2 seconds\n

**9.45.5.54 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

**9.45.5.55 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

#### 9.45.5.56 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

#### 9.45.5.57 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

#### 9.45.5.58 ULONG SLQSWdsSetEventReport ( wdsSetEventReportReq \* pSetEventReportReq )

This API sets the wireless data connection state reporting conditions for the requesting control point.

##### Parameters

<i>pSetEventReportReq[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

#### 9.45.5.59 ULONG SLQSWdsSwiPDPRuntimeSettings ( swiPDPRuntimeSettingsReq \* pPDPRuntimeSettingsReq, swiPDPRuntimeSettingsResp \* 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="#">swiPDPRuntimeSettingsReq</a> for more information</li> </ul>
<i>pPDPRuntimeSettingsResp</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">swiPDPRuntimeSettingsResp</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

#### 9.45.5.60 BOOL WDS\_IsGobiDevice ( )

## 9.46 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)

### 9.46.1 Enumeration Type Documentation

#### 9.46.1.1 enum eQMI\_NAS\_GET\_RF\_INFO\_RESP

Enumerator

**[eTLV\\_RF\\_BAND\\_INFO](#)**

### 9.46.2 Function Documentation

#### 9.46.2.1 enum eQCWWANError PkQmiNasGetRFBandInfo ( WORD \* pMlength, BYTE \* pBuffer )

#### 9.46.2.2 enum eQCWWANError UpkQmiNasGetRFBandInfo ( BYTE \* pMdmResp, struct QmiNasGetRFBandInfoResp \* pApiResp )

## 9.47 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)

## 9.47.1 Macro Definition Documentation

9.47.1.1 `#define FORBIDDEN_INDEX 4`

9.47.1.2 `#define INDEX_ZERO 0`

9.47.1.3 `#define MAX_DESCRIPTION_LENGTH 255`

9.47.1.4 `#define PREFERRED_INDEX 6`

9.47.1.5 `#define QMI_NAS_MAX_INSTANCES 20`

9.47.1.6 `#define QMI_NAS_NETSTATUS_MASK 0x03`

9.47.1.7 `#define ROAMING_INDEX 2`

## 9.47.2 Enumeration Type Documentation

9.47.2.1 `enum eQMI_NAS_PERFORM_NETWORK_SCAN_RESP`

Enumerator

***eTLV\_3GPP\_NETWORK\_INFO***

## 9.47.3 Function Documentation

9.47.3.1 `enum eQCWWANError PkQmiNasPerformNetworkScan ( WORD * pMlength, BYTE * pParamField )`

9.47.3.2 `enum eQCWWANError UpkQmiNasPerformNetworkScan ( BYTE * pMdmResp, struct QmiNasPerformNetworkScanResp * pAipResp )`

## 9.48 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\_INVALID\_XID,
  - eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED,
  - 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\_FREE\_PROFILE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE,
  - eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF,
  - eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE,

```

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 }

```

## 9.48.1 Enumeration Type Documentation

### 9.48.1.1 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\_INVALID\_XID** 36- Invalid transaction id  
**eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED** 37- Unsupported multiple SMS  
**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 intercase is not muticast  
**eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE** 1062 - Maximum requests in use  
**eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE** 1063 - Invalid muticast 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\_FW\_ENTER\_DOWNLOAD\_MODE** 0xE80A - Enter Download Mode

**eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE** 0xE80B - File transfer to modem complete

**eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT** 0xE80C - Wait for modem to reboot

**eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE** 0xE80D - Invalid Crash State for Firmware Download

**eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT\_ACTIVE\_IMAGE** 0xE80E - Same as current active image

**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

## 9.48.1.2 enum qm\_wds\_ds\_profile\_extended\_err\_codes

## WDS DS profile extended error codes

## Enumerator

- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL** 1 - General Failure
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HANDLE** 2 - The request contains an invalid profile handle
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP** 3 - An invalid operation was requested.
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE** 4 - The request contains an invalid technology type
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM** 5 - The request contains an invalid profile number
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT** 6 - The request contains an invalid profile identifier
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID** 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\_INVALID\_SUBS\_ID** 11 - The request contains an invalid subscription identifier.
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY** 12 - The request contains an invalid profile family.
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_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**

## 9.49 qos.h File Reference

## Data Structures

- struct [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t](#)
- struct [pack\\_qos\\_SLQSQosSmiReadApnExtraParams\\_t](#)
- struct [unpack\\_qos\\_SLQSQosSmiReadApnExtraParams\\_t](#)

- struct [pack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#)
- struct [unpack\\_QosFlowStat\\_t](#)
- struct [unpack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind\\_t](#)
- struct [pack\\_qos\\_SLQSSetQosEventCallback\\_t](#)
- struct [unpack\\_qos\\_QosFlowInfoState\\_t](#)
- struct [unpack\\_qos\\_dataRate\\_t](#)
- struct [unpack\\_qos\\_tokenBucket\\_t](#)
- struct [unpack\\_qos\\_pktErrRate\\_t](#)
- struct [unpack\\_qos\\_swiQosFlow\\_t](#)
- struct [unpack\\_qos\\_IPv4Addr\\_t](#)
- struct [unpack\\_qos\\_Tos\\_t](#)
- struct [unpack\\_qos\\_IPv6Addr\\_t](#)
- struct [unpack\\_qos\\_IPv6TrafCls\\_t](#)
- struct [unpack\\_qos\\_Port\\_t](#)
- struct [unpack\\_qos\\_swiQosFilter\\_t](#)
- struct [unpack\\_qos\\_QosFlowInfo\\_t](#)
- struct [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind\\_t](#)

## Macros

- `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`
- `#define LIBPACK_MAX_QOS_FILTERS 25`
- `#define LIBPACK_MAX_QOS_FLOWS 8`

## Functions

- int [pack\\_qos\\_SLQSQosGetNetworkStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_qos\\_SLQSQosGetNetworkStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSQosSwiReadApnExtraParams](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSQosSwiReadApnExtraParams](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSQosSwiReadDataStats](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSQosSwiReadDataStats](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSQosSwiReadDataStats\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosNWStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosPriEventCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_qos\\_SLQSSetQosEventCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_SLQSSetQosEventCallback\\_t](#) reqParam)
- int [unpack\\_qos\\_SLQSSetQosEventCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind\\_t](#) \*pOutput)

## 9.49.1 Macro Definition Documentation

9.49.1.1 `#define LIBPACK_MAX_QOS_FILTERS 25`

9.49.1.2 `#define LIBPACK_MAX_QOS_FLOW_PER_APN_STATS 10`

9.49.1.3 `#define LIBPACK_MAX_QOS_FLOWS 8`

## 9.49.2 Function Documentation

9.49.2.1 `int pack_qos_SLQSQosGetNetworkStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack command to retrieve QoS status of the network. This maps to SLQSQosGetNetworkStatus

### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</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
- Technology Supported: CDMA
- PDN Specific: No

9.49.2.2 `int pack_qos_SLQSQosSwiReadApnExtraParams ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSwiReadApnExtraParams_t reqParam )`

Function to pack QMI command to query extra APN parameters This maps to SLQSQosSwiReadApnExtraParams

### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>

<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_SLQSQosSwiReadApnExtraParams_t</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
- PDN Specific: Yes

**9.49.2.3** `int pack_qos_SLQSQosSwiReadDataStats ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSwiReadDataStats_t reqParam )`

Function to pack QMI command to query APN data statistics This maps to SLQSQosSwiReadDataStats

**Parameters**

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_SLQSQosSwiReadDataStats_t</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
  - PDN Specific: Yes

9.49.2.4 `int pack_qos_SLQSSetQosEventCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSSetQosEventCallback_t reqParam )`

Function to pack QMI command to enable QoS event indications This maps to SLQSSetQosEventCallback

## Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_SLQSSetQosEventCallback_t</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
  - PDN Specific: Yes

9.49.2.5 `int unpack_qos_SLQSQosGetNetworkStatus ( uint8_t * pResp, uint16_t respLen, unpack_qos_SLQSQosGetNetworkStatus_t * pOutput )`

Function to unpack the response to get NW QoS status command This maps to SLQSQosGetNetworkStatus

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSQosGetNetworkStatus_t</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

**9.49.2.6** `int unpack_qos_SLQSQoSswiReadApnExtraParams ( uint8_t * pResp, uint16_t respLen,  
unpack_qos_SLQSQoSswiReadApnExtraParams_t * pOutput )`

Function to unpack the response to get NW QoS status command This maps to SLQSQoSswiReadApnExtraParams

**Parameters**

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSQoSswiReadApnExtraParams_t</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

**9.49.2.7** `int unpack_qos_SLQSQoSswiReadDataStats ( uint8_t * pResp, uint16_t respLen,  
unpack_qos_SLQSQoSswiReadDataStats_t * pOutput )`

Function to unpack APN data statistics response This maps to SLQSQoSswiReadDataStats

**Parameters**

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSQoSswiReadDataStats_t</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

#### 9.49.2.8 int unpack\_qos\_SLQSSetQosEventCallback ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>

## Returns

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

## See Also

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

#### 9.49.2.9 int unpack\_qos\_SLQSSetQosEventCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_qos\_SLQSSetQosEventCallback\_ind\_t \* *pOutput* )

Function to unpack QoS event indications This maps to SLQSSetQosEventCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_qos_SLQSSetQosEventCallback_ind_t</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 is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_NETWORK\_STATUS\_IND indication to identify this event from QOS service read function

9.49.2.10 `int unpack_qos_SLQSSetQosNWStatusCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_qos_SLQSSetQosNWStatusCallback_ind_t * pOutput )`

Function to unpack QoS NW status indication. This maps to SLQSSetQosNWStatusCallback

#### Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosNWStatusCallback_ind_t</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
- This is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_NETWORK\_STATUS\_IND indication to identify this event from QOS service read function

9.49.2.11 `int unpack_qos_SLQSSetQosPriEventCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_qos_SLQSSetQosPriEventCallback_ind_t * pOutput )`

Function to unpack QoS primary flow events. This maps to SLQSSetQosPriEventCallback

#### Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosPriEventCallback_ind_t</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 is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_PRIMARY\_QOS\_EVENT\_IND indication to identify this event from QOS service read function
- This is only generated when the primary flow is modified by the host

9.49.2.12 `int unpack_qos_SLQSSetQosStatusCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_qos_SLQSSetQosStatusCallback_ind_t * pOutput )`

Function to unpack QoS status indications. This maps to SLQSSetQosStatusCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_SLQSSetQosStatusCallback_ind_t</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 is a broadcast notification that is sent to the registered QoS service
- No explicit function to register for this indication is needed
- Please use eQMI\_QOS\_FLOW\_STATUS\_IND indication to identify this event from QOS service read function

## 9.50 sms.h File Reference

### Data Structures

- struct [pack\\_sms\\_SLQSGetSMS\\_t](#)
- struct [unpack\\_sms\\_SLQSGetSMS\\_t](#)
- struct [pack\\_sms\\_SLQSGetSMSList\\_t](#)
- struct [qmiSmsMessageList](#)

- struct [unpack\\_sms\\_SLQSGetSMSList\\_t](#)
- struct [pack\\_sms\\_SLQSMModifySMSStatus\\_t](#)
- struct [unpack\\_sms\\_SLQSMModifySMSStatus\\_t](#)
- struct [pack\\_sms\\_SLQSDDeleteSMS\\_t](#)
- struct [unpack\\_sms\\_SLQSDDeleteSMS\\_t](#)
- struct [pack\\_sms\\_SendSMS\\_t](#)
- struct [unpack\\_sms\\_SendSMS\\_t](#)
- struct [pack\\_sms\\_SetNewSMSCallback\\_t](#)
- struct [unpack\\_sms\\_SetNewSMSCallback\\_t](#)
- struct [sMSMTMessage](#)
- struct [newMTMessageTlv](#)
- struct [sMSTransferRouteMTMessage](#)
- struct [transferRouteMessageTlv](#)
- struct [sMSMessageMode](#)
- struct [messageModeTlv](#)
- struct [sMSEtwsMessage](#)
- struct [sMSEtwsMessageTlv](#)
- struct [sMSEtwsPlmn](#)
- struct [eTWSPLMNInfoTlv](#)
- struct [sMSCAddress](#)
- struct [sMSCAddressTlv](#)
- struct [sMSOnIMS](#)
- struct [sMSOnIMSTlv](#)
- struct [unpack\\_sms\\_SetNewSMSCallback\\_ind\\_t](#)
- struct [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t](#)

## Macros

- #define [MAX\\_SMS\\_MESSAGE\\_SIZE](#) 2048
- #define [MAX\\_SMS\\_LIST\\_SIZE](#) 255
- #define [MAX\\_MS\\_TRANSFER\\_ROUTE\\_MSG](#) 256
- #define [MAX\\_MSE\\_TWS\\_MSG](#) 1254
- #define [MAX\\_MSC\\_ADDRESS\\_SIZE](#) 256
- #define [MAX\\_CDMA\\_ENC\\_MO\\_TXT\\_MSG\\_SIZE](#) 255

## Typedefs

- 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](#)

## Enumerations

- enum [eqmiCbkJSetStatus](#) {  
[LIBPACK\\_QMI\\_CBK\\_PARAM\\_RESET](#) = 0,  
[LIBPACK\\_QMI\\_CBK\\_PARAM\\_SET](#) = 1,  
[LIBPACK\\_QMI\\_CBK\\_PARAM\\_NOCHANGE](#) }

## Functions

- int [pack\\_sms\\_SLQSGetSMS](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSGetSMS\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSGetSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetSMS\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSGetSMSList](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSGetSMSList\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSGetSMSList](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetSMSList\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSModifySMSStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSModifySMSStatus\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSModifySMSStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSModifySMSStatus\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSDeleteSMS](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSDeleteSMS\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSDeleteSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSDeleteSMS\\_t](#) \*pOutput)
- int [pack\\_sms\\_SendSMS](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SendSMS\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SendSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SendSMS\\_t](#) \*pOutput)
- int [pack\\_sms\\_SetNewSMSCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SetNewSMSCallback\\_t](#) reqParam)
- int [unpack\\_sms\\_SetNewSMSCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SetNewSMSCallback\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SetNewSMSCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SetNewSMSCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t](#) \*pOutput)

## 9.50.1 Macro Definition Documentation

9.50.1.1 `#define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255`

9.50.1.2 `#define MAX_MS_TRANSFER_ROUTE_MSG 256`

9.50.1.3 `#define MAX_MSC_ADDRESS_SIZE 256`

9.50.1.4 `#define MAX_MSE_TWS_MSG 1254`

9.50.1.5 `#define MAX_SMS_LIST_SIZE 255`

9.50.1.6 `#define MAX_SMS_MESSAGE_SIZE 2048`

## 9.50.2 Typedef Documentation

9.50.2.1 `typedef struct sMSCAddress sMSCAddressInfo`

### 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>

9.50.2.2 typedef struct **sMSEtwsMessage** **sMSEtwsMessageInfo**

## 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>

9.50.2.3 typedef struct **sMSEtwsPlmn** **sMSEtwsPlmnInfo**

## 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>

9.50.2.4 typedef struct **sSMSMessageMode** **sSMSMessageModeInfo**

## Parameters

<i>messageMode</i>	Message Mode
--------------------	--------------

9.50.2.5 typedef struct **sMSMTMessage** **sMSMTMessageInfo**

## Parameters

<i>storageType</i>	memory storage 0x00-UIM 0x01-NV
<i>messageIndex</i>	MT Message index

9.50.2.6 typedef struct **sMSOnIMS** **sMSOnIMSInfo**

## Parameters

<i>smsOnIMS</i>	SMS on IMS
-----------------	------------

9.50.2.7 typedef struct **sMSTransferRouteMTMessage** **sMSTransferRouteMTMessageInfo**

## 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>

## 9.50.3 Enumeration Type Documentation

## 9.50.3.1 enum eqmiCbkJSetStatus

## Enumerator

**LIBPACK\_QMI\_CBK\_PARAM\_RESET**  
**LIBPACK\_QMI\_CBK\_PARAM\_SET**  
**LIBPACK\_QMI\_CBK\_PARAM\_NOCHANGE**

## 9.50.4 Function Documentation

9.50.4.1 `int pack_sms_SendSMS ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SendSMS_t * reqParam )`

send sms list pack

## Parameters

<i>in, out</i>	<i>pCtx</i>	qmi request context
<i>out</i>	<i>pReqBuf</i>	qmi request buffer
<i>out</i>	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

## Returns

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

## See Also

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

9.50.4.2 `int pack_sms_SetNewSMSCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SetNewSMSCallback_t reqParam )`

set new sms callback pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

**Returns**

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

**See Also**

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

9.50.4.3 int pack\_sms\_SLQSDeleteSMS ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_sms\_SLQSDeleteSMS\_t \* *reqParam* )

delete sms pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

**Returns**

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

**See Also**

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

9.50.4.4 int pack\_sms\_SLQSGetSMS ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_sms\_SLQSGetSMS\_t \* *reqParam* )

get sms pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

**Returns**

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

**See Also**

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

9.50.4.5 `int pack_sms_SLQSGetSMSList ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSGetSMSList_t * reqParam )`

get sms list pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

#### Returns

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

#### See Also

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

9.50.4.6 `int pack_sms_SLQSModifySMSStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSModifySMSStatus_t * reqParam )`

modify sms status pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
	<i>reqParam</i>	packed request

#### Returns

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

#### See Also

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

9.50.4.7 `int unpack_sms_SendSMS ( uint8_t * pResp, uint16_t respLen, unpack_sms_SendSMS_t * pOutput )`

send sms unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

## See Also

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

9.50.4.8 `int unpack_sms_SetNewSMSCallback ( uint8_t * pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_t * Output )`

set new sms callback unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.50.4.9 `int unpack_sms_SetNewSMSCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_sms_SetNewSMSCallback_ind_t * pOutput )`

set new sms callback indication unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.50.4.10 `int unpack_sms_SLQSDDeleteSMS ( uint8_t * pResp, uint16_t respLen, unpack_sms_SLQSDDeleteSMS_t * pOutput )`

delete sms unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.50.4.11 int unpack\_sms\_SLQSGetSMS ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSGetSMS\_t \* *pOutput* )

get sms unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.50.4.12 int unpack\_sms\_SLQSGetSMSList ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSGetSMSList\_t \* *pOutput* )

get sms list unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.50.4.13 int unpack\_sms\_SLQSModifySMSStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSModifySMSStatus\_t \* *pOutput* )

modify sms status unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

9.50.4.14 int unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t \* *pOutput* )

sms full callback indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

## 9.51 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](#)

### 9.51.1 Detailed Description

SWI data types.

### 9.51.2 Macro Definition Documentation

9.51.2.1 `#define QMI_NO_LTE_FW_SUPPORT 0`

9.51.2.2 `#define QMI_TLV_PLACEHOLDER 0x8F`

9.51.2.3 `#define SWI_API`

9.51.2.4 `#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.

### 9.51.3 Typedef Documentation

9.51.3.1 `typedef int BOOL`

9.51.3.2 `typedef unsigned char BYTE`

9.51.3.3 `typedef char CHAR`

9.51.3.4 `typedef float FLOAT`

9.51.3.5 `typedef signed int INT32`

9.51.3.6 `typedef signed char INT8`

9.51.3.7 `typedef const char* LPCSTR`

9.51.3.8 `typedef signed short SHORT`

9.51.3.9 `typedef unsigned long ULONG`

9.51.3.10 `typedef unsigned long long ULONGLONG`

9.51.3.11 `typedef unsigned short USHORT`

9.51.3.12 `typedef unsigned short WORD`

## 9.52 swiloc.h File Reference

### Data Structures

- struct [unpack\\_swiloc\\_SwiLocGetAutoStart\\_t](#)
- struct [pack\\_swiloc\\_SwiLocSetAutoStart\\_t](#)

### Functions

- int [pack\\_swiloc\\_SwiLocGetAutoStart](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)

- int [unpack\\_swiloc\\_SwiLocGetAutoStart](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiloc\\_SwiLocGetAutoStart\\_t](#) \*pOutput)
- int [pack\\_swiloc\\_SwiLocSetAutoStart](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiloc\\_SwiLocSetAutoStart\\_t](#) \*reqArg)
- int [unpack\\_swiloc\\_SwiLocSetAutoStart](#) (uint8\_t \*pResp, uint16\_t respLen)

### 9.52.1 Function Documentation

#### 9.52.1.1 int [pack\\_swiloc\\_SwiLocGetAutoStart](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Get Auto Start pack

##### Parameters

in	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

##### Returns

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

##### See Also

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

#### 9.52.1.2 int [pack\\_swiloc\\_SwiLocSetAutoStart](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_swiloc\\_SwiLocSetAutoStart\\_t](#) \* *reqArg* )

Set Auto Start pack

##### Parameters

in	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

##### Returns

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

##### See Also

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

#### 9.52.1.3 int [unpack\\_swiloc\\_SwiLocGetAutoStart](#) ( uint8\_t \* *pResp*, uint16\_t *respLen*, [unpack\\_swiloc\\_SwiLocGetAutoStart\\_t](#) \* *pOutput* )

Get Auto Start unpack

##### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

#### 9.52.1.4 int unpack\_swiloc\_SwiLocSetAutoStart ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Set Auto Start unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

## 9.53 swioma.h File Reference

**Data Structures**

- struct [pack\\_swioma\\_SLQSOMADMStartSession\\_t](#)
- struct [unpack\\_swioma\\_SLQSOMADMStartSession\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMCancelSession\\_t](#)
- struct [unpack\\_swioma\\_SLQSOMADMGetSettings\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMSetSettings\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMSelectSelection\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMGetSessionInfo\\_t](#)
- struct [unpack\\_swioma\\_SLQSOMADMGetSessionInfo\\_t](#)
- struct [unpack\\_omaDmFotaTlv\\_t](#)
- struct [unpack\\_omaDmConfigTlv\\_t](#)
- struct [unpack\\_omaDmNotificationsTlv\\_t](#)
- struct [unpack\\_swioma\\_SLQSOMADMAAlertCallback\\_ind\\_t](#)

**Macros**

- #define [LIBPACK\\_MAX\\_SWIOMA\\_STR\\_LEN](#) 255

**Functions**

- int [pack\\_swioma\\_SLQSOMADMStartSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swioma\\_SLQSOMADMStartSession\\_t](#) reqParam)
- int [unpack\\_swioma\\_SLQSOMADMStartSession](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMStartSession\\_t](#) \*pOutput)

- int [pack\\_swima\\_SLQSOMADMCancelSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swima\\_SLQSOMADMCancelSession\\_t](#) reqParam)
- int [unpack\\_swima\\_SLQSOMADMCancelSession](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swima\\_SLQSOMADMGetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swima\\_SLQSOMADMGetSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swima\\_SLQSOMADMGetSettings\\_t](#) \*pOutput)
- int [pack\\_swima\\_SLQSOMADMSetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swima\\_SLQSOMADMSetSettings\\_t](#) reqParam)
- int [unpack\\_swima\\_SLQSOMADMSetSettings](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swima\\_SLQSOMADMSendSelection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swima\\_SLQSOMADMSendSelection\\_t](#) reqParam)
- int [unpack\\_swima\\_SLQSOMADMSendSelection](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_swima\\_SLQSOMADMGetSessionInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swima\\_SLQSOMADMGetSessionInfo\\_t](#) reqParam)
- int [unpack\\_swima\\_SLQSOMADMGetSessionInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swima\\_SLQSOMADMGetSessionInfo\\_t](#) \*pOutput)
- int [pack\\_swima\\_SLQSOMADMAAlertCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swima\\_SLQSOMADMAAlertCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_swima\\_SLQSOMADMAAlertCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swima\\_SLQSOMADMAAlertCallback\\_ind\\_t](#) \*pOutput)

### 9.53.1 Macro Definition Documentation

9.53.1.1 `#define LIBPACK_MAX_SWIOMA_STR_LEN 255`

### 9.53.2 Function Documentation

9.53.2.1 `int pack_swima_SLQSOMADMAAlertCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

#### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</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

9.53.2.2 `int pack_swioma_SLQSOMADMCancelSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMCancelSession_t reqParam )`

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

## Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swioma_SLQSOMADMCancelSession_t</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

9.53.2.3 `int pack_swioma_SLQSOMADMGetSessionInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMGetSessionInfo_t reqParam )`

Function to pack QMI command to return information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

## Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>

<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_swima_SLQSOMADMGetSessionInfo_t</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

#### 9.53.2.4 int pack\_swima\_SLQSOMADMGetSettings ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Function to pack command to retrieve the OMA-DM settings from the device. This maps to SLQSOMADMGetSettings2

**Parameters**

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Buffer for packed QMI command to be provided by the host application</li> <li>• Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</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

9.53.2.5 `int pack_swioma_SLQSOMADMSendSelection ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMSendSelection_t reqParam )`

Function to pack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

#### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swioma_SLQSOMADMSendSelection_t</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

9.53.2.6 `int pack_swioma_SLQSOMADMSetSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMSetSettings_t reqParam )`

Function to pack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

#### Parameters

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swioma_SLQSOMADMSetSettings_t</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

**9.53.2.7** `int pack_swima_SLQSOMADMStartSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMStartSession_t reqParam )`

Function to pack Start OMA-DM session command This maps to SLQSOMADMStartSession2

**Parameters**

<i>pCtx</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>pReqBuf</i> [IN/OUT]	<ul style="list-style-type: none"> <li>Buffer for packed QMI command to be provided by the host application</li> <li>Minimum expected size is 2048 bytes</li> </ul>
<i>pLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
<i>reqParam</i> [IN]	<ul style="list-style-type: none"> <li>See <a href="#">pack_swima_SLQSOMADMStartSession_t</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

**9.53.2.8** `int unpack_swima_SLQSOMADMAAlertCallback ( uint8_t * pResp, uint16_t respLen )`

Function to unpack response of QMI command to enable the SWIOMADM network-initiated alert callback function. This maps to SetSLQSOMADMAAlertCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

## Returns

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

## See Also

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

## Note

- Please use eQMI\_SWIOMA\_EVENT\_IND indication to identify this event from SWIOMA service read function

**9.53.2.9** int unpack\_swioma\_SLQSOMADMAAlertCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_swioma\_SLQSOMADMAAlertCallback\_ind\_t \* *pOutput* )

Function to unpack SWIOMADM alert indications This maps to SetSLQSOMADMAAlertCallback

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swioma_SLQSOMADMAAlertCallback_ind_t</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

**9.53.2.10** int unpack\_swioma\_SLQSOMADMCancelSession ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSession

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

## Returns

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

## See Also

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

9.53.2.11 int unpack\_swima\_SLQSOMADMGetSessionInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swima\_SLQSOMADMGetSessionInfo\_t \* *pOutput* )

Function to unpack information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfo

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swima_SLQSOMADMGetSessionInfo_t</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

9.53.2.12 int unpack\_swima\_SLQSOMADMGetSettings ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swima\_SLQSOMADMGetSettings\_t \* *pOutput* )

Function to unpack OMA-DM get settings response from modem This maps to SLQSOMADMGetSettings2

## Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swioma_SLQSOMADMGetSettings_t</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

#### 9.53.2.13 int unpack\_swioma\_SLQSOMADMSendSelection ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelection2

**Parameters**

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

**Returns**

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

**See Also**

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

#### 9.53.2.14 int unpack\_swioma\_SLQSOMADMSetSettings ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

**Parameters**

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>

**Returns**

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

**See Also**

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

9.53.2.15 `int unpack_swima_SLQSOMADMStartSession ( uint8_t * pResp, uint16_t respLen,  
unpack_swima_SLQSOMADMStartSession_t * pOutput )`

Function to unpack Start OMA-DM session response from modem This maps to SLQSOMADMStartSession2

#### Parameters

<i>pResp</i> [IN]	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
<i>respLen</i> [IN]	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
<i>pOutput</i> [OUT]	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swima_SLQSOMADMStartSession_t</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

## 9.54 SWIWWANCMAPI.h File Reference

## 9.55 uim.h File Reference

### Data Structures

- struct [uim\\_appStatus](#)
- struct [uim\\_slotInfo](#)
- struct [uim\\_cardStatus](#)
- struct [uim\\_hotSwapStatus](#)
- struct [unpack\\_uim\\_GetCardStatus\\_t](#)
- struct [uim\\_encryptedPIN1](#)
- struct [uim\\_remainingRetries](#)
- struct [uim\\_sessionInformation](#)
- struct [uim\\_verifyUIMPIN](#)
- struct [uim\\_unblockUIMPIN](#)
- struct [uim\\_cardResult](#)
- struct [uim\\_setPINProtection](#)
- struct [uim\\_changeUIMPIN](#)
- struct [uim\\_fileInfo](#)
- struct [uim\\_UIMSessionInformation](#)
- struct [uim\\_readTransparentInfo](#)
- struct [uim\\_readResult](#)
- struct [pack\\_uim\\_VerifyPin\\_t](#)
- struct [unpack\\_uim\\_VerifyPin\\_t](#)
- struct [pack\\_uim\\_UnblockPin\\_t](#)
- struct [unpack\\_uim\\_UnblockPin\\_t](#)
- struct [pack\\_uim\\_SetPinProtection\\_t](#)

- struct [unpack\\_uim\\_SetPinProtection\\_t](#)
- struct [pack\\_uim\\_ChangePin\\_t](#)
- struct [unpack\\_uim\\_ChangePin\\_t](#)
- struct [pack\\_uim\\_ReadTransparent\\_t](#)
- struct [unpack\\_uim\\_ReadTransparent\\_t](#)
- struct [pack\\_uim\\_SLQSUIMEventRegister\\_t](#)
- struct [unpack\\_uim\\_SLQSUIMEventRegister\\_t](#)
- struct [appStats](#)
- struct [slotInf](#)
- struct [unpack\\_uim\\_SLQSUIMSetStatusChangeCallBack\\_ind\\_t](#)
- struct [slot\\_t](#)
- struct [slots\\_t](#)
- struct [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t](#)
- struct [pack\\_uim\\_SLQSUIMSwitchSlot\\_t](#)
- struct [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind\\_t](#)

## Macros

- [#define UIM\\_UINT8\\_MAX\\_STRING\\_SZ 255](#)
- [#define UIM\\_MAX\\_DESCRIPTION\\_LENGTH 255](#)
- [#define UIM\\_MAX\\_NO\\_OF\\_SLOTS 5](#)
- [#define UIM\\_MAX\\_NO\\_OF\\_APPLICATIONS 10](#)
- [#define MAX\\_NO\\_OF\\_SLOTS 5](#)
- [#define MAX\\_NO\\_OF\\_APPLICATIONS 10](#)
- [#define MAX\\_DESCRIPTION\\_LENGTH 255](#)
- [#define MAX\\_SLOTS\\_STATUS 255](#)
- [#define MAX\\_ICCID\\_LENGTH 255](#)

## Functions

- int [pack\\_uim\\_GetCardStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_uim\\_GetCardStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_GetCardStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_VerifyPin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_VerifyPin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_VerifyPin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_VerifyPin\\_t](#) \*pOutput)
- int [pack\\_uim\\_UnblockPin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_UnblockPin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_UnblockPin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_UnblockPin\\_t](#) \*pOutput)
- int [pack\\_uim\\_SetPinProtection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SetPinProtection\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SetPinProtection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SetPinProtection\\_t](#) \*pOutput)
- int [pack\\_uim\\_ChangePin](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_ChangePin\\_t](#) \*reqArg)
- int [unpack\\_uim\\_ChangePin](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_ChangePin\\_t](#) \*pOutput)
- int [pack\\_uim\\_ReadTransparent](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_ReadTransparent\\_t](#) \*reqArg)
- int [unpack\\_uim\\_ReadTransparent](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_ReadTransparent\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMEventRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMEventRegister\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMEventRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMEventRegister\\_t](#) \*pOutput)

- int [unpack\\_uim\\_SLQSUIMSetStatusChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMSetStatusChangeCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMGetSlotsStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_uim\\_SLQSUIMGetSlotsStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMSwitchSlot](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMSwitchSlot\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMSwitchSlot](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind\\_t](#) \*pOutput)

### 9.55.1 Macro Definition Documentation

9.55.1.1 `#define MAX_DESCRIPTION_LENGTH 255`

9.55.1.2 `#define MAX_ICCID_LENGTH 255`

9.55.1.3 `#define MAX_NO_OF_APPLICATIONS 10`

9.55.1.4 `#define MAX_NO_OF_SLOTS 5`

9.55.1.5 `#define MAX_SLOTS_STATUS 255`

9.55.1.6 `#define UIM_MAX_DESCRIPTION_LENGTH 255`

9.55.1.7 `#define UIM_MAX_NO_OF_APPLICATIONS 10`

9.55.1.8 `#define UIM_MAX_NO_OF_SLOTS 5`

9.55.1.9 `#define UIM_UINT8_MAX_STRING_SZ 255`

### 9.55.2 Function Documentation

9.55.2.1 int [pack\\_uim\\_ChangePin](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_uim\\_ChangePin\\_t](#) \* *reqArg* )

Change Pin pack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.55.2.2 int [pack\\_uim\\_GetCardStatus](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Get Card Status pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.55.2.3 int pack\_uim\_ReadTransparent ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_uim\_ReadTransparent\_t \* *reqArg* )

SLQS ReadTransparent pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.55.2.4 int pack\_uim\_SetPinProtection ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_uim\_SetPinProtection\_t \* *reqArg* )

Set Pin Protection pack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

## Returns

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

## See Also

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

9.55.2.5 `int pack_uim_SLQSUIMEventRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMEventRegister_t * reqArg )`

UIM Status Change callback enable pack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.55.2.6 `int pack_uim_SLQSUIGetSlotsStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

9.55.2.7 `int pack_uim_SLQSUISSwitchSlot ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUISSwitchSlot_t * reqArg )`

switch slot pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.55.2.8 `int pack_uim_UnblockPin ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_UnblockPin_t * reqArg )`

Unblock Pin pack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.55.2.9 `int pack_uim_VerifyPin ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_VerifyPin_t * reqArg )`

Verify Pin Status pack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response
in	<i>reqArg</i>	request parameter

#### Returns

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

#### See Also

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

9.55.2.10 `int unpack_uim_ChangePin ( uint8_t * pResp, uint16_t respLen, unpack_uim_ChangePin_t * pOutput )`

Change Pin unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

## See Also

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

9.55.2.11 `int unpack_uim_GetCardStatus ( uint8_t * pResp, uint16_t respLen, unpack_uim_GetCardStatus_t * pOutput )`

Get Card Status unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.55.2.12 `int unpack_uim_ReadTransparent ( uint8_t * pResp, uint16_t respLen, unpack_uim_ReadTransparent_t * pOutput )`

SLQS ReadTransparent unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

## Returns

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

## See Also

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

9.55.2.13 `int unpack_uim_SetPinProtection ( uint8_t * pResp, uint16_t respLen, unpack_uim_SetPinProtection_t * pOutput )`

Set Pin Protection unpack

## Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.55.2.14** int unpack\_uim\_SetUimSlotStatusChangeCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t \* *pOutput* )

UIM Slot Status Change indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**Note**

use pack\_uim\_SLQSUIEventRegister to subscribe

**9.55.2.15** int unpack\_uim\_SLQSUIEventRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIEvent-  
Register\_t \* *pOutput* )

UIM Status Change callback enable unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**9.55.2.16** int unpack\_uim\_SLQSUIGetSlotsStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIGetSlotsStatus\_t \* *pOutput* )

get slot status unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

9.55.2.17 int unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind\_t \* *pOutput* )

UIM Status Change indication unpack

**Parameters**

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

**Returns**

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

**See Also**

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

**Note**

use pack\_uim\_SLQSUIMEventRegister to subscribe

9.55.2.18 int unpack\_uim\_SLQSUIMSwitchSlot ( uint8\_t \* *pResp*, uint16\_t *respLen* )

switch slot unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

9.55.2.19 `int unpack_uim_UnblockPin ( uint8_t * pResp, uint16_t respLen, unpack_uim_UnblockPin_t * pOutput )`

Unblock Pin unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.55.2.20 `int unpack_uim_VerifyPin ( uint8_t * pResp, uint16_t respLen, unpack_uim_VerifyPin_t * pOutput )`

Verify Pin unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

## 9.56 wds.h File Reference

### Data Structures

- struct [LibPackQosClassID](#)
- struct [LibPackTFTIDParams](#)
- struct [LibPackGPRSRequestedQoS](#)
- struct [LibPackUMTSQoS](#)
- struct [LibPackUMTSReqQoSSigInd](#)
- struct [pack\\_wds\\_SLQSSStartDataSession\\_t](#)
- struct [unpack\\_wds\\_SLQSSStartDataSession\\_t](#)
- struct [unpack\\_wds\\_SLQSSetPacketSrvStatusCallback\\_t](#)
- struct [pack\\_wds\\_SLQSSStopDataSession\\_t](#)
- struct [wds\\_ProfileIdentifier](#)
- struct [wds\\_GPRSQoS](#)
- struct [wds\\_PCSCFIPv4ServerAddressList](#)
- struct [wds\\_PCSCFFQDNAddress](#)
- struct [wds\\_PCSCFFQDNAddressList](#)

- struct [wds\\_Domain](#)
- struct [wds\\_DomainNameList](#)
- struct [wds\\_IPV6AddressInfo](#)
- struct [wds\\_IPV6GWAddressInfo](#)
- struct [unpack\\_wds\\_SLQSGetRuntimeSettings\\_t](#)
- struct [wds\\_currNetworkInfo](#)
- struct [unpack\\_wds\\_SLQSSetWdsEventCallback\\_ind\\_t](#)
- struct [pack\\_wds\\_SLQSSetWdsEventCallback\\_t](#)
- struct [pack\\_wds\\_SLQSGetRuntimeSettings\\_t](#)
- struct [wds\\_UMTSMInQoS](#)
- struct [LibPackprofile\\_3GPP](#)
- struct [LibPackprofile\\_3GPP2](#)
- union [wds\\_profileInfo](#)
- struct [pack\\_wds\\_SLQSCreateProfile\\_t](#)
- struct [PackCreateProfileOut](#)
- struct [unpack\\_wds\\_SLQSCreateProfile\\_t](#)
- struct [pack\\_wds\\_SLQSModifyProfile\\_t](#)
- struct [unpack\\_wds\\_SLQSModifyProfile\\_t](#)
- struct [pack\\_wds\\_SLQSGetProfileSettings\\_t](#)
- struct [LibpackProfile3GPP](#)
- struct [LibpackProfile3GPP2](#)
- union [unpackWdsProfileParam](#)
- struct [UnPackGetProfileSettingOut](#)
- struct [unpack\\_wds\\_SLQSGetProfileSettings\\_t](#)
- struct [unpack\\_wds\\_GetSessionState\\_t](#)
- struct [pack\\_wds\\_GetDefaultProfile\\_t](#)
- struct [unpack\\_wds\\_GetDefaultProfile\\_t](#)
- struct [unpack\\_wds\\_GetConnectionRate\\_t](#)
- struct [pack\\_wds\\_GetPacketStatus\\_t](#)
- struct [unpack\\_wds\\_GetPacketStatus\\_t](#)
- struct [unpack\\_wds\\_GetSessionDuration\\_t](#)
- struct [pack\\_wds\\_GetSessionDuration\\_t](#)
- struct [unpack\\_wds\\_GetDormancyState\\_t](#)
- struct [pack\\_wds\\_GetDormancyState\\_t](#)
- struct [pack\\_wds\\_SLQSDeleteProfile\\_t](#)
- struct [unpack\\_wds\\_SLQSDeleteProfile\\_t](#)
- struct [pack\\_wds\\_SetDefaultProfile\\_t](#)
- struct [unpack\\_wds\\_SLQSGet3GPPConfigItem\\_t](#)
- struct [pack\\_wds\\_SLQSSet3GPPConfigItem\\_t](#)
- struct [unpack\\_wds\\_GetMobileIP\\_t](#)
- struct [pack\\_wds\\_GetMobileIP\\_t](#)
- struct [pack\\_wds\\_GetMobileIPProfile\\_t](#)
- struct [unpack\\_wds\\_GetMobileIPProfile\\_t](#)
- struct [currNetworkInfo](#)
- struct [unpack\\_wds\\_SLQSGetCurrDataSystemStat\\_t](#)
- struct [pack\\_wds\\_SLQSGetCurrDataSystemStat\\_t](#)
- struct [unpack\\_wds\\_GetLastMobileIPError\\_t](#)
- struct [pack\\_wds\\_GetLastMobileIPError\\_t](#)
- struct [rmTrasnferStaticsReq](#)
- struct [pack\\_wds\\_RMSetTransferStatistics\\_t](#)
- struct [unpack\\_wds\\_RMSetTransferStatistics\\_t](#)
- struct [pack\\_wds\\_SetMobileIPProfile\\_t](#)
- struct [unpack\\_wds\\_SetMobileIPProfile\\_t](#)
- struct [pack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_t](#)
- struct [ipv6AddressInfo](#)

- struct [unpack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_t](#)
- struct [transferStatInd](#)
- struct [pack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [connectionStatus](#)
- struct [dunchannelRate](#)
- struct [unpack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [qmiWDSDataBearerTechnology](#)
- struct [unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#)
- struct [pack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#)
- struct [pack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#)
- struct [unpack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#)
- struct [pack\\_wds\\_SetDefaultProfileNum\\_t](#)
- struct [pack\\_wds\\_GetDefaultProfileNum\\_t](#)
- struct [unpack\\_wds\\_GetDefaultProfileNum\\_t](#)
- struct [wdsDhcpv4ProfileId](#)
- struct [wdsDhcpv4HwConfig](#)
- struct [wdsDhcpv4Option](#)
- struct [wdsDhcpv4OptionList](#)
- struct [pack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#)
- struct [unpack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#)

## Macros

- [#define IPV6\\_ADDRESS\\_ARRAY\\_SIZE 8](#)
- [#define MAX\\_WDS\\_3GPP\\_CONF\\_LTE\\_ATTACH\\_PROFILE\\_LIST\\_SIZE 24](#)
- [#define PACK\\_WDS\\_IPV4 4](#)
- [#define PACK\\_WDS\\_IPV6 6](#)

## Typedefs

- typedef union [unpackWdsProfileParam](#) [UnpackQmiProfileInfo](#)

## Functions

- int [pack\\_wds\\_SLQSStartDataSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSStartDataSession\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSStartDataSession](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSStartDataSession\\_t](#) \*pOutput)
- int [unpack\\_wds\\_SLQSSetPacketSrvStatusCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetPacketSrvStatusCallback\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSStopDataSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSStopDataSession\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSStopDataSession](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_wds\\_SLQSGetRuntimeSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGetRuntimeSettings\\_t](#) \*pOutput)
- int [unpack\\_wds\\_SLQSSetWdsEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetWdsEventCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_wds\\_SLQSSetWdsEventCallback](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_SLQSSetWdsEventCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetWdsEventCallback\\_t](#) \*reqArg)
- int [pack\\_wds\\_SLQSGetRuntimeSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSGetRuntimeSettings\\_t](#) \*reqArg)
- int [pack\\_wds\\_SLQSCreateProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSCreateProfile\\_t](#) \*reqArg)

- [int unpack\\_wds\\_SLQSCreateProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSCreateProfile\\_t](#) \*pOutput)
- [int pack\\_wds\\_SLQSModifyProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSModifyProfile\\_t](#) \*reqArg)
- [int unpack\\_wds\\_SLQSModifyProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSModifyProfile\\_t](#) \*pOutput)
- [int pack\\_wds\\_SLQSGetProfileSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSGetProfileSettings\\_t](#) \*reqArg)
- [int unpack\\_wds\\_SLQSGetProfileSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGetProfileSettings\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetSessionState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_wds\\_GetSessionState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetSessionState\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetDefaultProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetDefaultProfile\\_t](#) \*reqParam)
- [int unpack\\_wds\\_GetDefaultProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetDefaultProfile\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetConnectionRate](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_wds\\_GetConnectionRate](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetConnectionRate\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetPacketStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetPacketStatus\\_t](#) \*reqParam)
- [int unpack\\_wds\\_GetPacketStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetPacketStatus\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetSessionDuration](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetSessionDuration\\_t](#) \*reqParam)
- [int unpack\\_wds\\_GetSessionDuration](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetSessionDuration\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetDormancyState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetDormancyState\\_t](#) \*reqParam)
- [int unpack\\_wds\\_GetDormancyState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetDormancyState\\_t](#) \*pOutput)
- [int pack\\_wds\\_SLQSDeleteProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSDeleteProfile\\_t](#) \*reqParam)
- [int unpack\\_wds\\_SLQSDeleteProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSDeleteProfile\\_t](#) \*pOutput)
- [int pack\\_wds\\_SetDefaultProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetDefaultProfile\\_t](#) \*reqParam)
- [int unpack\\_wds\\_SetDefaultProfile](#) (uint8\_t \*pResp, uint16\_t respLen)
- [int pack\\_wds\\_SLQSGet3GPPConfigItem](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_wds\\_SLQSGet3GPPConfigItem](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGet3GPPConfigItem\\_t](#) \*pOutput)
- [int pack\\_wds\\_SLQSSet3GPPConfigItem](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSet3GPPConfigItem\\_t](#) \*reqParam)
- [int unpack\\_wds\\_SLQSSet3GPPConfigItem](#) (uint8\_t \*pResp, uint16\_t respLen)
- [int pack\\_wds\\_GetMobileIP](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetMobileIP\\_t](#) \*pReqParam)
- [int unpack\\_wds\\_GetMobileIP](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetMobileIP\\_t](#) \*pOutput)
- [int pack\\_wds\\_GetMobileIPProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetMobileIPProfile\\_t](#) \*reqParam)
- [int unpack\\_wds\\_GetMobileIPProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetMobileIPProfile\\_t](#) \*pOutput)
- [int pack\\_wds\\_SLQSGetCurrDataSystemStat](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSGetCurrDataSystemStat\\_t](#) \*pReqParam)
- [int unpack\\_wds\\_SLQSGetCurrDataSystemStat](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGetCurrDataSystemStat\\_t](#) \*pOutput)

- int [pack\\_wds\\_GetLastMobileIPError](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetLastMobileIPError\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_GetLastMobileIPError](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetLastMobileIPError\\_t](#) \*pOutput)
- int [pack\\_wds\\_RMSetTransferStatistics](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_RMSetTransferStatistics\\_t](#) \*reqParam)
- int [unpack\\_wds\\_RMSetTransferStatistics](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_RMSetTransferStatistics\\_t](#) \*pOutput)
- int [pack\\_wds\\_SetMobileIPProfile](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetMobileIPProfile\\_t](#) \*reqParam)
- int [unpack\\_wds\\_SetMobileIPProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SetMobileIPProfile\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSWdsSviPDPRuntimeSettings](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSWdsSviPDPRuntimeSettings\\_t](#) \*reqParam)
- int [unpack\\_wds\\_SLQSWdsSviPDPRuntimeSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSWdsSviPDPRuntimeSettings\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSGetDUNCallInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSGetDUNCallInfo\\_t](#) \*reqParam)
- int [unpack\\_wds\\_SLQSGetDUNCallInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGetDUNCallInfo\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSGetDataBearerTechnology](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SLQSGetDataBearerTechnology](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSGetDataBearerTechnology\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetIPFamilyPreference](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SLQSSetIPFamilyPreference](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetIPFamilyPreference\\_t](#) \*pOutput)
- int [pack\\_wds\\_SetDefaultProfileNum](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SetDefaultProfileNum\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_SetDefaultProfileNum](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_wds\\_GetDefaultProfileNum](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetDefaultProfileNum\\_t](#) \*pReqParam)
- int [unpack\\_wds\\_GetDefaultProfileNum](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetDefaultProfileNum\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSetDHCPv4ClientConfig](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#) \*pReq)
- int [unpack\\_wds\\_SLQSSetDHCPv4ClientConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#) \*pOutput)

### 9.56.1 Macro Definition Documentation

9.56.1.1 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

9.56.1.2 `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`

9.56.1.3 `#define PACK_WDS_IPV4 4`

9.56.1.4 `#define PACK_WDS_IPV6 6`

### 9.56.2 Typedef Documentation

9.56.2.1 `typedef union unpackWdsProfileParam UnpackQmiProfileInfo`

### 9.56.3 Function Documentation

9.56.3.1 `int pack_wds_GetConnectionRate ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get connection rate pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: No

9.56.3.2 `int pack_wds_GetDefaultProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfile_t * reqParam )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.3 `int pack_wds_GetDefaultProfileNum ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfileNum_t * pReqParam )`

get default profile number pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**9.56.3.4** `int pack_wds_GetDormancyState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDormancyState_t * reqParam )`

get dormancy state pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**Note**

PDN Specific: No

**9.56.3.5** `int pack_wds_GetLastMobileIPError ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetLastMobileIPError_t * pReqParam )`

get current data system pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**Note**

PDN Specific: No

9.56.3.6 `int pack_wds_GetMobileIP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetMobileIP_t * pReqParam )`

get mobile ip mode pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: No

9.56.3.7 `int pack_wds_GetMobileIPProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetMobileIPProfile_t * reqParam )`

get mobile ip profile pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.8 `int pack_wds_GetPacketStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetPacketStatus_t * reqParam )`

get packet status pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**Note**

PDN Specific: Yes

9.56.3.9 `int pack_wds_GetSessionDuration ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetSessionDuration_t * reqParam )`

get session duration pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**Note**

PDN Specific: No

9.56.3.10 `int pack_wds_GetSessionState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get session state pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

## Note

PDN Specific: Yes

```
9.56.3.11 int pack_wds_RMSetTransferStatistics ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
        pack_wds_RMSetTransferStatistics_t * reqParam )
```

rm set transfer statistics pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

## Note

PDN Specific: No

```
9.56.3.12 int pack_wds_SetDefaultProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
        pack_wds_SetDefaultProfile_t * reqParam )
```

set default profile pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

```
9.56.3.13 int pack_wds_SetDefaultProfileNum ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
        pack_wds_SetDefaultProfileNum_t * pReqParam )
```

set default profile number pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**9.56.3.14** int pack\_wds\_SetMobileIPProfile ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_wds\_SetMobileIPProfile\_t \* reqParam )

set mobile ip profile pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**Note**

PDN Specific: Yes

**9.56.3.15** int pack\_wds\_SLQSCreateProfile ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_wds\_SLQSCreateProfile\_t \* reqArg )

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**Note**

PDN Specific: Yes

9.56.3.16 `int pack_wds_SLQSDeleteProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSDeleteProfile_t * reqParam )`

delete stored profile pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.17 `int pack_wds_SLQSGet3GPPConfigItem ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get 3Gpp config items pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.18 `int pack_wds_SLQSGetCurrDataSystemStat ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSGetCurrDataSystemStat_t * pReqParam )`

get current data system pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.19 int pack\_wds\_SLQSGetDataBearerTechnology ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSGetDataBearerTechnology\_t \* *pReqParam* )

get data bearer technology pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.20 int pack\_wds\_SLQSGetDUNCallInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSGetDUNCallInfo\_t \* *reqParam* )

get dun call info pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.21 `int pack_wds_SLQSGetProfileSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSGetProfileSettings_t * reqArg )`

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.22 `int pack_wds_SLQSGetRuntimeSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSGetRuntimeSettings_t * reqArg )`

get runtime settings pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.23 `int pack_wds_SLQSModifyProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSModifyProfile_t * reqArg )`

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.24 `int pack_wds_SLQSSet3GPPConfigItem ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSet3GPPConfigItem_t * reqParam )`

set 3Gpp config items pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### Returns

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

#### See Also

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

#### Note

PDN Specific: Yes

9.56.3.25 `int pack_wds_SLQSSetIPFamilyPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSetIPFamilyPreference_t * pReqParam )`

Set IP Family Preference pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**9.56.3.26** int pack\_wds\_SLQSSetWdsEventCallback ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSSetWdsEventCallback\_t \* *reqArg* )

set event callback pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

**Returns**

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

**See Also**

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

**Note**

PDN Specific: No

**9.56.3.27** int pack\_wds\_SLQSSGetDHCIPv4ClientConfig ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSSGetDHCIPv4ClientConfig\_t \* *pReq* )

get DHCPv4 Client Config pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReq</i>	request parameter

**Returns**

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

**See Also**

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

**9.56.3.28** int pack\_wds\_SLQSStartDataSession ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSStartDataSession\_t \* *reqArg* )

Start data session

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.29 int pack\_wds\_SLQSStopDataSession ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSStopDataSession\_t \* *reqArg* )

stop data session pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

## Returns

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

## See Also

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

## Note

PDN Specific: Yes

9.56.3.30 int pack\_wds\_SLQSWdsSwiPDPRuntimeSettings ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t \* *reqParam* )

swi pdp runtime settings pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReq</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqParam</i>	request parameter

**Returns**

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

**See Also**

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

**Note**

PDN Specific: Yes

**9.56.3.31** int unpack\_wds\_GetConnectionRate ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetConnectionRate\_t \* *pOutput* )

get connection rate unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.56.3.32** int unpack\_wds\_GetDefaultProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetDefaultProfile\_t \* *pOutput* )

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.56.3.33** int unpack\_wds\_GetDefaultProfileNum ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetDefaultProfile-Num\_t \* *pOutput* )

get default profile number unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.56.3.34 int unpack\_wds\_GetDormancyState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetDormancyState\_t \* *pOutput* )

get dormancy state unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.56.3.35 int unpack\_wds\_GetLastMobileIPError ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetLastMobileIPError\_t \* *pOutput* )

get current data system unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.56.3.36 int unpack\_wds\_GetMobileIP ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetMobileIP\_t \* *pOutput* )

get mobile ip mode unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

9.56.3.37 int unpack\_wds\_GetMobileIPProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetMobileIPProfile\_t \* *pOutput* )

get mobile ip profile unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

9.56.3.38 int unpack\_wds\_GetPacketStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetPacketStatus\_t \* *pOutput* )

get packet status unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

9.56.3.39 `int unpack_wds_GetSessionDuration ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetSessionDuration_t * pOutput )`

get session duration unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.40 `int unpack_wds_GetSessionState ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetSessionState_t * pOutput )`

get session state unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.41 `int unpack_wds_RMSetTransferStatistics ( uint8_t * pResp, uint16_t respLen, unpack_wds_RMSetTransferStatistics_t * pOutput )`

rm set transfer statistics unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

#### Returns

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

#### See Also

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

#### 9.56.3.42 int unpack\_wds\_SetDefaultProfile ( uint8\_t \* *pResp*, uint16\_t *respLen* )

set default profile unpack

##### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

##### Returns

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

##### See Also

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

#### 9.56.3.43 int unpack\_wds\_SetDefaultProfileNum ( uint8\_t \* *pResp*, uint16\_t *respLen* )

set default profile number unpack

##### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

##### Returns

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

##### See Also

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

#### 9.56.3.44 int unpack\_wds\_SetMobileIPProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetMobileIPProfile\_t \* *pOutput* )

set mobile ip profile unpack

##### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

##### Returns

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

##### See Also

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

9.56.3.45 `int unpack_wds_SLQSCreateProfile ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSCreateProfile_t * pOutput )`

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response structure to fill
in	<i>pProfileId</i>	profile id pointer passed in req

#### Returns

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

#### See Also

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

9.56.3.46 `int unpack_wds_SLQSDeleteProfile ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSDeleteProfile_t * pOutput )`

delete stored profile unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.47 `int unpack_wds_SLQSGet3GPPConfigItem ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSGet3GPPConfigItem_t * pOutput )`

get 3GPP config items unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.48 `int unpack_wds_SLQSGetCurrDataSystemStat ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_SLQSGetCurrDataSystemStat_t * pOutput )`

get current data system unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.49 `int unpack_wds_SLQSGetDataBearerTechnology ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_SLQSGetDataBearerTechnology_t * pOutput )`

get data bearer technology unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.50 `int unpack_wds_SLQSGetDUNCallInfo ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSGetDUNCall-  
Info_t * pOutput )`

get dun call info unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

## See Also

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

9.56.3.51 `int unpack_wds_SLQSGetProfileSettings ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettings_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response structure to fill

## Returns

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

## See Also

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

9.56.3.52 `int unpack_wds_SLQSGetRuntimeSettings ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSGetRuntimeSettings_t * pOutput )`

get runtime settings unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.56.3.53 `int unpack_wds_SLQSModifyProfile ( uint8_t * pResp, uint16_t respLen, unpack_wds_SLQSModifyProfile_t * pOutput )`

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.56.3.54 `int unpack_wds_SLQSSet3GPPConfigItem ( uint8_t * pResp, uint16_t respLen )`

set 3GPP config items unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

9.56.3.55 `int unpack_wds_SLQSSetIPFamilyPreference ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_SLQSSetIPFamilyPreference_t * pOutput )`

Set IP Family Preference unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

#### Returns

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

#### See Also

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

9.56.3.56 `int unpack_wds_SLQSSetPacketSrvStatusCallback ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_SLQSSetPacketSrvStatusCallback_t * pOutput )`

set packet srv status callback unpack

#### Parameters

in	<i>pResp</i>	qmi response
in	<i>respLen</i>	length
out	<i>pOutput</i>	unpacked response

#### Returns

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

#### See Also

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

9.56.3.57 `int unpack_wds_SLQSSetWdsEventCallback ( uint8_t * pResp, uint16_t respLen )`

set event callback unpack

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

**Returns**

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

**See Also**

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

**9.56.3.58** int unpack\_wds\_SLQSSetWdsEventCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t \* *pOutput* )

set event callback unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.56.3.59** int unpack\_wds\_SLQSSGetDhCpV4ClientConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSGetDhCpV4ClientConfig\_t \* *pOutput* )

get DHCPv4 Client Config unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**Returns**

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

**See Also**

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

**9.56.3.60** int unpack\_wds\_SLQSStartDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSStartData-  
Session\_t \* *pOutput* )

start data session unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

9.56.3.61 int unpack\_wds\_SLQSStopDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen* )

stop data session unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length

## Returns

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

## See Also

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

9.56.3.62 int unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t \* *pOutput* )

get current data system unpack

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

## Returns

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

## See Also

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

# Index

- [\\_3gppRelease](#)
  - [pack\\_wds\\_SLQSSet3GPPConfigItem\\_t, 576](#)
  - [unpack\\_wds\\_SLQSGet3GPPConfigItem\\_t, 962](#)
- [\\_GetProfileSettingIn, 56](#)
  - [ProfileID, 56](#)
  - [ProfileType, 56](#)
- [\\_GetProfileSettingOut, 56](#)
  - [curProfile, 57](#)
  - [pExtErrCode, 57](#)
- [\\_MitigationDevInfo, 59](#)
  - [deviceId, 59](#)
  - [deviceIdLen, 59](#)
- [\\_SLQSOMADMSessionInfo, 68](#)
  - [pDate, 70](#)
  - [pDateLength, 70](#)
  - [pPkgDescLength, 70](#)
  - [pPkgDescription, 70](#)
  - [pPkgName, 70](#)
  - [pPkgNameLength, 70](#)
  - [pRetryCount, 70](#)
  - [pSessionState, 70](#)
  - [pSessionType, 70](#)
  - [pSeverity, 70](#)
  - [pSource, 70](#)
  - [pSourceLength, 70](#)
  - [pStatus, 70](#)
  - [pTime, 70](#)
  - [pTimeLength, 70](#)
  - [pUpdateCompleteStatus, 71](#)
- [\\_SLQSOMADMSettings, 71](#)
  - [pAutosdm, 72](#)
  - [pFOTAUpdate, 72](#)
  - [pFOTAdownload, 72](#)
  - [pFwAutoCheck, 72](#)
  - [pOMADMEEnabled, 72](#)
- [\\_SLQSOMADMSettingsReqParams, 72](#)
  - [FOTAUpdate, 73](#)
  - [FOTAdownload, 73](#)
  - [pAutosdm, 73](#)
- [\\_SLQSOMADMSettingsReqParams3, 73](#)
  - [FOTAUpdate, 74](#)
  - [FOTAdownload, 74](#)
  - [pAutosdm, 74](#)
  - [pFwAutoCheck, 74](#)
- [\\_SLQSSwiGetHostDevInfoParams, 74](#)
  - [bManSize, 75](#)
  - [bModelSize, 75](#)
  - [bPlasmaIDSize, 75](#)
  - [bSWVerSize, 75](#)
  - [pManString, 75](#)
  - [pModelString, 75](#)
  - [pPlasmaIDString, 75](#)
  - [pSWVerString, 75](#)
- [\\_SLQSSwiGetOSInfoParams, 75](#)
  - [bNameSize, 76](#)
  - [bVersionSize, 76](#)
  - [pNameString, 76](#)
  - [pVersionString, 76](#)
- [\\_SLQSSwiGetSerialNoExtParams, 76](#)
  - [meidLength, 77](#)
  - [pMeidString, 77](#)
- [\\_SLQSSwiSetHostDevInfoParams, 77](#)
  - [bManSize, 78](#)
  - [bModelSize, 78](#)
  - [bPlasmaIDSize, 78](#)
  - [bSWVerSize, 78](#)
  - [pManString, 78](#)
  - [pModelString, 78](#)
  - [pPlasmaIDString, 78](#)
  - [pSWVerString, 78](#)
- [\\_SLQSSwiSetOSInfoParams, 78](#)
  - [bNameSize, 79](#)
  - [bVersionSize, 79](#)
  - [pNameString, 79](#)
  - [pVersionString, 79](#)
- [\\_SlqsNas3GppNetworkRAT\\_, 66](#)
  - [MCC, 66](#)
  - [MNC, 66](#)
  - [RAT, 67](#)
- [\\_getIndicationRegResp, 55](#)
  - [pRegCallStatInfoEvt, 56](#)
  - [pRegTransLayerInfoEvt, 56](#)
  - [pRegTransNWRRegInfoEvt, 56](#)
- [\\_getResetInfoNotification, 57](#)
  - [source, 58](#)
  - [type, 58](#)
- [\\_getTransLayerInfoResp, 58](#)
  - [pRegInd, 58](#)
  - [pTransLayerInfo, 58](#)
- [\\_getTransNWRRegInfoResp, 58](#)
  - [pRegStatus, 59](#)
- [\\_modemTempNotification, 59](#)
  - [ModemTempState, 60](#)
  - [ModemTemperature, 60](#)
- [\\_packetSrvStatus, 60](#)
  - [bearerID, 62](#)
  - [connStatus, 62](#)
  - [ipFamily, 62](#)

- pQmiInterfaceInfo, 62
- reconfigReqd, 62
- sessionEndReason, 62
- techName, 62
- verboseSessnEndReason, 62
- verboseSessnEndReasonType, 62
- \_qaQmi3GPP2BroadcastCfgInfo, 62
  - activated\_ind, 63
  - CDMABroadcastConfig, 63
  - num\_instances, 63
- \_qaQmi3GPPBroadcastCfgInfo, 63
  - activated\_ind, 63
  - broadcastConfig, 63
  - num\_instances, 63
- \_setIndicationRegReq, 63
  - pRegCallStatInfoEvt, 64
  - pRegTransLayerInfoEvt, 64
  - pRegTransNWRegInfoEvt, 64
- \_slqs3GPPConfigItem, 64
  - LTEAttachProfileListLen, 66
  - p3gppRelease, 66
  - pDefaultPDNEnabled, 66
  - pLTEAttachProfile, 66
  - pLTEAttachProfileList, 66
  - pProfileList, 66
- \_slqsNetworkScanInfo, 67
  - pNetworkInfo, 68
  - pNetworkInfoInstances, 68
  - pPCSDigitInfo, 68
  - pPCSDigitInstances, 68
  - pRATInfo, 68
  - pRATInstances, 68
  - pScanResult, 68
- \_sysSelectPrefInfo, 79
  - pBandPref, 84
  - pEmerMode, 84
  - pGWAcqOrderPref, 84
  - pLTETBandPref, 84
  - pModePref, 84
  - pNetSelPref, 84
  - pPRLPref, 84
  - pRoamPref, 84
  - pSrvDomainPref, 84
- \_sysSelectPrefParams, 85
  - pAcqOrderPref, 90
  - pBandPref, 90
  - pCSGID, 90
  - pChgDuration, 90
  - pEmerMode, 90
  - pGWAcqOrderPref, 90
  - pLTETBandPref, 90
  - pMNCIncPCSDigStat, 90
  - pModePref, 90
  - pNetSelPref, 90
  - pPRLPref, 90
  - pRAT, 90
  - pRoamPref, 90
  - pSrvDomainPref, 91
  - pSrvRegRestriction, 91
  - pTdsdmaBandPref, 91
  - \_transLayerInfoNotification, 91
    - pTransLayerInfo, 92
    - regInd, 92
  - \_transLayerinfo, 91
    - TransCap, 91
    - TransType, 91
  - \_transNWRegInfoNotification, 92
    - NWRegStat, 92
- AAASPI
  - unpack\_wds\_GetMobileIPProfile\_t, 958
- AAASState
  - unpack\_wds\_GetMobileIPProfile\_t, 958
- ABSOLUTE\_VALIDITY
  - qaGobiApiSms.h, 1393
- ALS
  - getAllCallInformation, 229
- AMSSString
  - unpack\_dms\_GetFirmwareRevision\_t, 849
  - unpack\_dms\_GetFirmwareRevisions\_t, 850
- APNName
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 966
- absoluteValidity
  - cdmaMsgDecodingParams, 145
- accelAcceptReady
  - qaGobiApiCbK.h, 1166
- accelAcceptReady\_s, 93
  - batchPerSec, 93
  - injectEnable, 93
  - samplesPerBatch, 93
- accelTempAcceptReady
  - qaGobiApiCbK.h, 1166
- accelTempAcceptReady\_s, 93
  - batchPerSec, 94
  - injectEnable, 94
  - samplesPerBatch, 94
- AccessMac
  - protocolSubtypeElement, 613
- accolc
  - pack\_nas\_SetACCOLC\_t, 533
- ackIndicator
  - SMSTransferRouteMTMessage, 746
  - sMSTransferRouteMTMessage, 745
- acqOrdeLen
  - acqOrderPref, 95
  - nas\_acqOrderPref, 385
- acqOrderPref, 94
  - acqOrdeLen, 95
  - pAcqOrder, 95
- acroamsetting
  - slqsautoconnect, 715
- acsetting
  - slqsautoconnect, 715
- ActPilotPNElement, 95
  - ActSetPilotPN, 95
  - ActSetPilotPNStrength, 95
- ActSetCnt

- NetworkStat1x, 506
- ActSetPilotPN
  - ActPilotPNElement, 95
- ActSetPilotPNStrength
  - ActPilotPNElement, 95
- action
  - slqsautoconnect, 715
  - ssdatasession\_params, 751
- ActivateAutomatic
  - qaGobiApiDms.h, 1254
- activated\_ind
  - \_qaQmi3GPP2BroadcastCfgInfo, 63
  - \_qaQmi3GPPBroadcastCfgInfo, 63
- activationStatus
  - dms\_ActivationStatusTlv, 200
- ActivationStatusTlv
  - unpack\_dms\_SetEventReport\_ind\_t, 856
- activeBandClass
  - nas\_RFInfoTlv, 442
  - RFBandInfoElements, 661
- activeChannel
  - nas\_RFInfoTlv, 442
  - RFBandInfoElements, 661
- activeInd
  - messageWaitingInfoContent, 380
- ActiveStatus
  - CLIPResp, 158
  - CLIRResp, 158
  - CNAPResp, 160
  - COLPResp, 161
  - COLRResp, 162
- ActiveTechPref
  - unpack\_nas\_GetNetworkPreference\_t, 887
- AddCDMASysInfo, 95
  - geoSysIdx, 96
  - regPrd, 96
- AddSysInfo, 96
  - cellBroadcastCap, 96
  - geoSysIdx, 96
- addr
  - IPv4Addr, 304
  - IPv6Addr, 304
  - unpack\_qos\_IPv4Addr\_t, 919
  - unpack\_qos\_IPv6Addr\_t, 919
- address
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- aid
  - uim\_sessionInformation, 805
  - uim\_UIMSessionInformation, 808
  - UIMRefreshEvent, 823
  - UIMSessionInformation, 826
- aidLength
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 797
  - uim\_sessionInformation, 805
  - uim\_UIMSessionInformation, 808
  - UIMRefreshEvent, 823
- UIMSessionInformation, 826
- aidVal
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- aidingIndicatorMask
  - loc\_sensorDataUsage, 343
  - sensorDataUsage\_s, 676
  - t\_sensor, 775
- airTimer, 96
  - airTimerValue, 97
  - namID, 97
- airTimerValue
  - airTimer, 97
- alertPitch
  - signalInfo, 710
- alertingPattern
  - arrAlertingPattern, 107
- AlertingType
  - arrAlertingType, 108
- alertmsg
  - omaDmConfigTlv, 512
  - omaDmConfigTlvExt, 514
  - unpack\_omaDmConfigTlv\_t, 915
- alertmsglength
  - omaDmConfigTlv, 512
  - omaDmConfigTlvExt, 514
  - unpack\_omaDmConfigTlv\_t, 915
- allCallsAlphaIDInfo, 97
  - AlphaIDInfo, 97
  - callID, 97
- allCallsAlphaIDInfoArr
  - arrAlphaID, 108
- allCallsDiagInfo, 97
  - callID, 98
  - DiagInfo, 98
- AllCallsUUSInfo
  - arrUUSInfo, 114
- allCallsUUSInfo, 98
  - callID, 98
  - uusInfo, 98
- alphaDcs
  - alphaIDInfo, 99
- AlphaID
  - CatAlPhalIdentifierTlv, 137
- AlphaIDInfo
  - allCallsAlphaIDInfo, 97
- alphaIDInfo, 98
  - alphaDcs, 99
  - alphaLen, 99
  - alphaText, 99
- alphaIDLen
  - SMSAsyncRawSend\_s, 732
- AlphaIDLength
  - CatAlPhalIdentifierTlv, 137
- alphaLen
  - alphaIDInfo, 99
- alphaText

- alphaIDInfo, 99
- alphabet
  - wcdmaMsgEncodingParams, 1037
- Altitude
  - GPSSStateInfo, 265
- altitudeSrcInfo, 99
  - coverage, 100
  - linkage, 100
  - source, 100
- ambr\_dl
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- ambr\_dl\_ext
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- ambr\_dl\_ext2
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- ambr\_ul
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- ambr\_ul\_ext
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- ambr\_ul\_ext2
  - sApnExtraParams, 672
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
- amssSize
  - unpack\_dms\_GetFirmwareRevision\_t, 849
  - unpack\_dms\_GetFirmwareRevisions\_t, 850
- AnswerUSSD
  - qaGobiApiVoice.h, 1486
- apdoxypages.c, 1069
- apnId
  - pack\_qos\_SLQSQosSwiReadApnExtraParams\_t, 550
  - pack\_qos\_SLQSQosSwiReadDataStats\_t, 550
  - sApnExtraParams, 672
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 924
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- apnName
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-\_t, 972
- apnname
  - unpack\_wds\_GetDefaultProfile\_t, 956
- apnsize
  - unpack\_wds\_GetDefaultProfile\_t, 956
- appNameLength
  - loc\_LocApplicationInfo, 342
  - LocApplicationInfo, 346
- appProviderLength
  - loc\_LocApplicationInfo, 342
  - LocApplicationInfo, 347
- appState
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- appStats, 100
  - aidLength, 103
  - aidVal, 103
  - appState, 103
  - appType, 103
  - persoFeature, 103
  - persoRetries, 103
  - persoState, 103
  - persoUnblockRetries, 103
  - pin1Retries, 103
  - pin1State, 103
  - pin2Retries, 103
  - pin2State, 103
  - puk1Retries, 103
  - puk2Retries, 103
  - univPin, 103
- AppStatus
  - slotInf, 712
  - slotInfo, 714
  - uim\_slotInfo, 807
- appStatus, 103
  - aidLength, 106
  - aidVal, 106
  - appState, 106
  - appType, 106
  - persoFeature, 106
  - persoRetries, 106
  - persoState, 106
  - persoUnblockRetries, 106
  - pin1Retries, 106
  - pin1State, 106
  - pin2Retries, 106
  - pin2State, 106
  - puk1Retries, 106
  - puk2Retries, 106
  - univPin, 106
- appType
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- appVersionLength
  - loc\_LocApplicationInfo, 342
  - LocApplicationInfo, 347
- appVersionValid
  - loc\_LocApplicationInfo, 342
  - LocApplicationInfo, 347
- Application
  - unpack\_nas\_GetCDMANetworkParameters\_t, 885
- appversion\_str
  - slqsfwinfo\_s, 717
  - unpack\_dms\_GetFirmwareInfo\_t, 849

- arfcn
  - GERANInfo, 227
  - gsmCellInfo, 267
  - nas\_GERANInfo, 404
  - nas\_gsmCellInfo, 406
- arrAlertingPattern, 106
  - alertingPattern, 107
  - callID, 107
  - numInstances, 107
- arrAlertingType, 107
  - AlertingType, 108
  - callID, 108
  - numInstances, 108
- arrAlphaID, 108
  - allCallsAlphaIDInfoArr, 108
  - numInstances, 108
- arrCallEndReason, 109
  - callEndReason, 110
  - callID, 110
  - numInstances, 110
- arrCallInfo, 110
  - getAllCallInfo, 110
  - numInstances, 110
- arrCallInformation
  - voiceSetAllCallStatusCbInfo, 1017
- arrCalledPartyNum, 108
  - CalledPartyNum, 109
  - numInstances, 109
- arrConnectPartyNum, 110
  - ConnectedPartyNum, 111
  - numInstances, 111
- arrDiagInfo, 111
  - DiagInfo, 111
  - numInstances, 111
- arrRedirPartyNum, 111
  - numInstances, 112
  - RedirPartyNum, 112
- arrRemotePartyName, 112
  - GetAllCallRmtPtyName, 112
  - numInstances, 112
- arrRemotePartyNum, 112
  - numInstances, 113
  - RmtPtyNum, 113
- arrSvcOption, 113
  - callID, 113
  - numInstances, 113
  - srvOption, 113
- arrUUSInfo, 114
  - AllCallsUUSInfo, 114
  - numInstances, 114
- arrfileInfo
  - registerRefresh, 657
  - UIMRefreshEvent, 823
- AtCmdPort
  - DcsUsbPortNames, 193
- Audio Service (AUDIO), 47
- auth
  - unpack\_wds\_GetDefaultProfile\_t, 956
- authData
  - UIMAuthenticateReq, 811
- AuthProt
  - protocolSubtypeElement, 613
- authenticateResult, 114
  - content, 115
  - contentLen, 115
- Authentication
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 966
- authentication
  - pack\_wds\_SetDefaultProfile\_t, 569
- authenticationData, 115
  - context, 115
  - data, 116
  - dataLen, 116
- Autosdm
  - unpack\_swima\_SLQSOMADMGetSettings\_t, 948
- avgPeriod
  - LTESigRptCfg, 371
  - LTESigRptConfig, 372
  - nas\_LTESigRptConfig, 426
- azimuth
  - satelliteInfo, 674
- bAltitudeAssumed
  - gnssSvInfoNotification, 260
- bEnable
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t, 544
- bForceDownload
  - pack\_fms\_SetImagesPreference\_t, 526
- bICCID
  - slot\_t, 711
  - UIMSlotStatus, 828
- bICCIDLength
  - slot\_t, 711
  - UIMSlotStatus, 828
- bLogicalSlot
  - pack\_uim\_SLQSUIMSwitchSlot\_t, 564
  - slot\_t, 711
  - UIMSlotStatus, 828
  - UIMSwitchSlotReq, 830
- bManSize
  - \_SLQSSwiGetHostDevInfoParams, 75
  - \_SLQSSwiSetHostDevInfoParams, 78
- bModelSize
  - \_SLQSSwiGetHostDevInfoParams, 75
  - \_SLQSSwiSetHostDevInfoParams, 78
- bNameSize
  - \_SLQSSwiGetOSInfoParams, 76
  - \_SLQSSwiSetOSInfoParams, 79
- bNumberOfPhySlots
  - UIMSlotStatusChangeInfo, 829
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, 952
- BOOL
  - SwiDataTypes.h, 1571
- BPTiv
  - NASQmiCbKnasSystemSelPrefInd, 492

- bPlasmaIDSize
  - \_SLQSSwiGetHostDevInfoParams, 75
  - \_SLQSSwiSetHostDevInfoParams, 78
- bResetStatistics
  - rmTrasferStaticsReq, 661
  - swiRMTrasferStaticsReq, 773
- bSWVerSize
  - \_SLQSSwiGetHostDevInfoParams, 75
  - \_SLQSSwiSetHostDevInfoParams, 78
- BUILD\_ID\_LEN
  - qaGobiApiFms.h, 1286
- bVersionSize
  - \_SLQSSwiGetOSInfoParams, 76
  - \_SLQSSwiSetOSInfoParams, 79
- BYTE
  - SwiDataTypes.h, 1571
- band
  - LTEInfo, 363
  - nas\_LTEInfo, 419
- band1900
  - gsmCellInfo, 267
  - nas\_gsmCellInfo, 406
- band\_pref
  - NASBandPreferenceTlv, 467
- BandCapability
  - unpack\_dms\_GetBandCapability\_t, 844
- bandCapability
  - BandCapabilityResp, 119
  - unpack\_dms\_SLQSSetBandCapability\_t, 862
- BandCapabilityResp, 116
  - bandCapability, 119
  - pLteBandCapability, 119
  - pTdsBandCapability, 119
- bandwidth
  - LTEInfo, 363
  - nas\_LTEInfo, 419
- baselId
  - CDMAInfo, 143
  - CDMASysInfo, 152
  - nas\_CDMAInfo, 389
  - nas\_CDMASysInfo, 393
- baseLat
  - CDMAInfo, 143
  - CDMASysInfo, 152
  - nas\_CDMAInfo, 389
  - nas\_CDMASysInfo, 393
- baseLong
  - CDMAInfo, 143
  - CDMASysInfo, 152
  - nas\_CDMAInfo, 389
  - nas\_CDMASysInfo, 393
- BasestationID
  - qaQmiServingSystemParam, 618
  - unpack\_nas\_SLQSSetServingSystem\_t, 896
- BasestationLatitude
  - qaQmiServingSystemParam, 618
  - unpack\_nas\_SLQSSetServingSystem\_t, 897
- BasestationLongitude
  - qaQmiServingSystemParam, 618
  - unpack\_nas\_SLQSSetServingSystem\_t, 897
- batchPerSec
  - accelAcceptReady\_s, 93
  - accelTempAcceptReady\_s, 94
  - gyroAcceptReady\_s, 272
  - gyroTempAcceptReady\_s, 273
- BdsSV, 119
  - id, 119
  - mask, 119
- BdsSVInfo, 119
  - len, 120
  - pSV, 120
- BearerID
  - unpack\_qos\_QosFlowInfo\_t, 922
- bearerID
  - \_packetSrvStatus, 62
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 968
- bearerId
  - sQosFlowStat, 747
  - unpack\_QosFlowStat\_t, 938
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 972
- bootSize
  - unpack\_dms\_GetFirmwareRevisions\_t, 850
- BootString
  - unpack\_dms\_GetFirmwareRevisions\_t, 850
- bootversion\_str
  - slqsfwinfo\_s, 717
  - unpack\_dms\_GetFirmwareInfo\_t, 849
- Broadcast
  - unpack\_nas\_GetCDMANetworkParameters\_t, 885
- BroadcastConfig, 120
  - fromServiceId, 121
  - selected, 121
  - toServiceId, 121
- broadcastConfig
  - \_qaQmi3GPPBroadcastCfgInfo, 63
- bsInfoValid
  - CDMASysInfo, 152
  - nas\_CDMASysInfo, 393
- bsPRev
  - CDMASysInfo, 152
  - nas\_CDMASysInfo, 393
- bsPRevValid
  - CDMASysInfo, 152
  - nas\_CDMASysInfo, 393
- bsic
  - GERANInfo, 227
  - nas\_GERANInfo, 404
- bsicId
  - gsmCellInfo, 267
  - nas\_gsmCellInfo, 406
- bucketSz
  - tokenBucket, 788
  - unpack\_qos\_tokenBucket\_t, 936
- buildID

- CurrImageInfo, [176](#)
  - FMSImageIdElement, [223](#)
  - image\_info\_t, [283](#)
  - ImageIdElement, [285](#)
- buildIDLen
  - CurrImageInfo, [176](#)
  - image\_info\_t, [283](#)
- buildIDLength
  - FMSImageIdElement, [223](#)
  - ImageIdElement, [285](#)
- buildId
  - FMSImageElement, [222](#)
  - ImageElement, [284](#)
- buildIdLength
  - FMSImageElement, [222](#)
  - ImageElement, [284](#)
- BurstDTMFInfo
  - voiceBurstDTMFInfo, [982](#)
- burstDTMFInfo, [121](#)
  - digitCnt, [121](#)
  - pCallID, [121](#)
  - pDigitBuff, [121](#)
- ByteLoopbackMode
  - WDSGetLoopbackData, [1061](#)
- ByteLoopbackMultiplier
  - WDSGetLoopbackData, [1061](#)
- ByteTotalsElmntsV4
  - WdsByteTotals, [1051](#)
- ByteTotalsElmntsV6
  - WdsByteTotals, [1051](#)
- CATEventDataType, [138](#)
  - eventMask, [138](#)
  - pErrorMask, [138](#)
- CATSendEnvelopeCommand
  - qaGobiApiCat.h, [1155](#)
- CATSendTerminalResponse
  - qaGobiApiCat.h, [1156](#)
- CBK\_DISABLE\_EVENT
  - qaGobiApiCbK.h, [1164](#)
- CBK\_ENABLE\_EVENT
  - qaGobiApiCbK.h, [1164](#)
- CBK\_NOCHANGE
  - qaGobiApiCbK.h, [1164](#)
- CCETlv
  - QmiCbKCatEventStatusReportInd, [619](#)
- CDMA\_P\_Rev
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- CDMABroadcastConfig, [140](#)
  - \_qaQmi3GPP2BroadcastCfgInfo, [63](#)
  - language, [141](#)
  - selected, [141](#)
  - serviceCategory, [141](#)
- CDMAChannel, [141](#)
  - priChA, [141](#)
  - priChB, [141](#)
  - secChA, [141](#)
  - secChB, [142](#)
- CDMAECIOThresh, [142](#)
  - CDMAECIOThreshListLen, [142](#)
  - pCDMAECIOThreshList, [142](#)
- CDMAECIOThreshListLen
  - CDMAECIOThresh, [142](#)
  - nas\_CDMAECIOThresh, [388](#)
- CDMAInfo, [142](#)
  - baseId, [143](#)
  - baseLat, [143](#)
  - baseLong, [143](#)
  - nid, [143](#)
  - refpn, [143](#)
  - sid, [143](#)
- CDMARSSIOThresh, [148](#)
  - CDMARSSIOThreshListLen, [148](#)
  - pCDMARSSIOThreshList, [148](#)
- CDMARSSIOThreshListLen
  - CDMARSSIOThresh, [148](#)
  - nas\_CDMARSSIOThresh, [390](#)
- CDMASSInfo, [148](#)
  - ecio, [149](#)
  - rssI, [149](#)
  - unpack\_nas\_SLQSNasGetSigInfo\_t, [907](#)
- CDMASysInfo, [149](#)
  - baseId, [152](#)
  - baseLat, [152](#)
  - baseLong, [152](#)
  - bsInfoValid, [152](#)
  - bsPRev, [152](#)
  - bsPRevValid, [152](#)
  - ccsSupported, [152](#)
  - ccsSupportedValid, [152](#)
  - cdmaSysIdValid, [152](#)
  - isSysPriMatch, [152](#)
  - isSysPriMatchValid, [152](#)
  - MCC, [152](#)
  - MNC, [152](#)
  - networkID, [152](#)
  - networkIdValid, [152](#)
  - pRevInUse, [153](#)
  - pRevInUseValid, [153](#)
  - packetZone, [153](#)
  - packetZoneValid, [153](#)
  - sysInfoCDMA, [153](#)
  - systemID, [153](#)
- CDMASysInfoExt, [153](#)
  - imsi\_11\_12, [153](#)
  - MCC, [153](#)
- CDMASystemInfoExt
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- CHAR
  - SwiDataTypes.h, [1571](#)
- CLIPResp, [157](#)
  - ActiveStatus, [158](#)
  - ProvisionStatus, [158](#)
- CLIRResp, [158](#)
  - ActiveStatus, [158](#)

- ProvisionStatus, 158
- CNAPResp, 160
  - ActiveStatus, 160
  - ProvisionStatus, 160
- COLPResp, 160
  - ActiveStatus, 161
  - ProvisionStatus, 161
- COLRResp, 161
  - ActiveStatus, 162
  - ProvisionStatus, 162
- CONFIG\_LEN
  - qaGobiApiSms.h, 1393
- CQIValueCW0
  - LteCQIParm, 359
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 911
- CQIValueCW1
  - LteCQIParm, 359
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 911
- CSDomain
  - unpack\_nas\_GetServingNetwork\_t, 888
- CSGID, 169
  - id, 170
  - mcc, 170
  - mnc, 170
  - mncPcsDigits, 170
  - rat, 170
- CUGIndex
  - CUGInfo, 171
- CUGInfo, 170
  - CUGIndex, 171
  - SuppOA, 171
  - SuppPrefCUG, 171
- CallBackK registration (CBK), 35
- CallBarStatus
  - qaQmiServingSystemParam, 618
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- callBarStatus, 122
  - csBarStatus, 123
  - psBarStatus, 123
- CallBarringSysInfo, 121
  - csBarStatus, 122
  - psBarStatus, 122
- callDuration
  - unpack\_wds\_GetSessionDuration\_t, 960
- CallEndReason
  - DUNCallInfoInd, 209
- callEndReason
  - arrCallEndReason, 110
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 964
- CallFWExtInfo
  - getCallFWExtInfo, 236
- callFWExtInfo, 127
  - noReplyTimer, 129
  - numLen, 129
  - numPlan, 129
  - numType, 129
  - number, 129
  - PI, 129
  - SI, 129
  - SvcClass, 129
  - SvcStatus, 129
- CallFWInfo
  - getCallFWInfo, 237
- callFWInfo, 129
  - noReplyTimer, 130
  - numLen, 130
  - number, 130
  - SvcClass, 130
  - SvcStatus, 130
- callFwdTypeAndPlan, 126
  - numberPlan, 127
  - numberType, 127
- callIID
  - allCallsAlphaIDInfo, 97
  - allCallsDiagInfo, 98
  - allCallsUUSInfo, 98
  - arrAlertingPattern, 107
  - arrAlertingType, 108
  - arrCallEndReason, 110
  - arrSvcOption, 113
  - callInfo, 131
  - DTMFInfo, 208
  - getAllCallRmtPtyName, 230
  - getAllCallRmtPtyNum, 230
  - peerNumberInfo, 587
  - voiceCallInfoReq, 983
  - voiceInfoRec, 1011
  - voiceOTASPStatusInfo, 1014
  - voicePrivacyInfo, 1015
  - voiceStopContDTMFInfo, 1027
  - voiceSUPSNotification, 1031
- callInfo, 130
  - callID, 131
  - callState, 131
  - callType, 132
  - direction, 132
  - mode, 132
- callNumber
  - voiceCallRequestParams, 987
- callState
  - callInfo, 131
- callType
  - callInfo, 132
- calledPartyInfo, 123
  - numLen, 124
  - numPlan, 124
  - numType, 124
  - number, 124
  - PI, 125
  - SI, 125
- CalledPartyNum
  - arrCalledPartyNum, 109
- calledPartySubAdd, 125
  - extBit, 125
  - oddEvenInd, 125
  - subAddr, 125

- subAddrLen, [125](#)
  - subAddrType, [125](#)
- callerID
  - callerIDInfo, [126](#)
  - connectNumInfo, [166](#)
- callerIDInfo, [126](#)
  - callerID, [126](#)
  - callerIDLen, [126](#)
  - PI, [126](#)
- callerIDLen
  - callerIDInfo, [126](#)
  - connectNumInfo, [166](#)
- callerName
  - remotePartyName, [658](#)
- Callinfo
  - getAllCallInformation, [229](#)
- callingPartyInfo, [132](#)
  - numLen, [133](#)
  - numPlan, [133](#)
  - numType, [133](#)
  - number, [133](#)
  - PI, [133](#)
  - SI, [133](#)
- CancelUSSD
  - qaGobiApiVoice.h, [1486](#)
- Card Application Toolkit (CAT), [38](#)
- cardResult, [133](#)
  - sw1, [134](#)
  - sw2, [134](#)
- cardState
  - slotInf, [712](#)
  - slotInfo, [714](#)
  - uim\_slotInfo, [807](#)
- cardStatus, [134](#)
  - index1xPri, [135](#)
  - index1xSec, [135](#)
  - indexGwPri, [135](#)
  - indexGwSec, [135](#)
  - numSlot, [135](#)
  - SlotInfo, [135](#)
- Carrier
  - fwinfo\_s, [226](#)
- carrier
  - CurrentImgList, [174](#)
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [864](#)
- carrier\_str
  - slqsfwinfo\_s, [717](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [849](#)
- CarrierImage\_t, [135](#)
  - m\_FwBuildId, [136](#)
  - m\_FwImageld, [136](#)
  - m\_PriBuildId, [136](#)
  - m\_PrImageld, [136](#)
  - m\_nCarrierId, [136](#)
  - m\_nFolderId, [136](#)
  - m\_nStorage, [136](#)
- CatAlPhalIdentifierTlv, [136](#)
  - AlphaID, [137](#)
  - AlphaIDLength, [137](#)
  - ReferenceID, [137](#)
- CatAlphaldtfr
  - currentCatEvent, [173](#)
- CatCommonEventTlv, [137](#)
  - CatEvent, [137](#)
  - EventID, [137](#)
  - EventLength, [137](#)
  - TlvPresent, [137](#)
- CatEndPS
  - currentCatEvent, [173](#)
- CatEndProactiveSessionTlv, [137](#)
  - EndProactiveSession, [138](#)
- CatEviDData
  - currentCatEvent, [173](#)
- CatEvent
  - CatCommonEventTlv, [137](#)
- CatEventIDDDataTlv, [138](#)
  - Data, [138](#)
  - DataLength, [138](#)
  - ReferenceID, [138](#)
- CatEventListTlv, [138](#)
  - SetupEventList, [139](#)
- CatEventLst
  - currentCatEvent, [173](#)
- CatRefresh
  - currentCatEvent, [173](#)
- CatRefreshTlv, [139](#)
  - RefreshMode, [139](#)
  - RefreshStage, [139](#)
- causeCode
  - SMSAsyncRawSend\_s, [732](#)
- ccSUPSType, [139](#)
  - reason, [140](#)
  - svcType, [140](#)
- ccsSupported
  - CDMASysInfo, [152](#)
  - nas\_CDMASysInfo, [393](#)
- ccsSupportedValid
  - CDMASysInfo, [152](#)
  - nas\_CDMASysInfo, [393](#)
- cdmaMsgDecodingParams, [143](#)
  - absoluteValidity, [145](#)
  - mcTimeStamp, [145](#)
  - messageLength, [145](#)
  - pAlertPriority, [145](#)
  - pCallbkAddr, [145](#)
  - pCallbkAddrLength, [145](#)
  - pDisplayMode, [145](#)
  - pLanguage, [145](#)
  - pMessage, [146](#)
  - pMessageID, [146](#)
  - pPriority, [146](#)
  - pPrivacy, [146](#)
  - pReadAcknowledgementReq, [146](#)
  - pRelativeValidity, [146](#)
  - pSenderAddr, [146](#)
  - pSenderAddrLength, [146](#)

- pTextMsg, [146](#)
  - pTextMsgLength, [146](#)
  - pUserAcknowledgementReq, [146](#)
- cdmaMsgEncodingParams, [146](#)
  - messageId, [147](#)
  - pCallbackAddr, [147](#)
  - pDestAddr, [147](#)
  - pEncodingAlphabet, [147](#)
  - pMessage, [147](#)
  - pMessageSize, [147](#)
  - pPriority, [147](#)
  - pRelValidity, [147](#)
  - pTextMsg, [148](#)
  - textMsgLength, [148](#)
- cdmaSSInfo, [149](#)
  - ecio, [149](#)
  - rsi, [149](#)
- cdmaSysIdValid
  - CDMASysInfo, [152](#)
  - nas\_CDMASysInfo, [393](#)
- cell\_resel\_priority
  - infoInterFreq, [302](#)
  - nas\_infoInterFreq, [416](#)
- cellBroadcastCap
  - AddSysInfo, [96](#)
  - nas\_AddSysInfo, [386](#)
- CellDb, [153](#)
  - mask, [154](#)
- CellID
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetservingSystem\_t, [897](#)
- cellID
  - GERANInfo, [227](#)
  - nas\_GERANInfo, [404](#)
  - nas\_UMTSInfo, [458](#)
  - UMTSInfo, [833](#)
- cellId
  - GSMSysInfo, [271](#)
  - LTESysInfo, [377](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [429](#)
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1040](#)
- cellIdValid
  - gsmCellInfo, [267](#)
  - GSMSysInfo, [271](#)
  - LTESysInfo, [377](#)
  - nas\_gsmCellInfo, [406](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [429](#)
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1041](#)
- cellInterFreqParams
  - infoInterFreq, [302](#)
  - nas\_infoInterFreq, [416](#)
- cellsTDD
  - nas\_umtsLTENbrCell, [460](#)
  - umtsLTENbrCell, [835](#)
- CellParams
  - LTEInfoIntraFreq, [365](#)
  - nas\_LTEInfoIntraFreq, [422](#)
- cellParams, [154](#)
  - pci, [155](#)
  - rsrp, [155](#)
  - rsrq, [155](#)
  - rsi, [155](#)
  - srxlev, [155](#)
- cellReselPriority
  - lteGsmCellInfo, [361](#)
  - LTEInfoIntraFreq, [365](#)
  - lteWcdmaCellInfo, [378](#)
  - nas\_lteGsmCellInfo, [417](#)
  - nas\_LTEInfoIntraFreq, [422](#)
  - nas\_lteWcdmaCellInfo, [431](#)
- cells\_len
  - infoInterFreq, [302](#)
  - lteGsmCellInfo, [361](#)
  - nas\_infoInterFreq, [416](#)
  - nas\_lteGsmCellInfo, [417](#)
- cellsLen
  - LTEInfoIntraFreq, [365](#)
  - lteWcdmaCellInfo, [378](#)
  - nas\_LTEInfoIntraFreq, [422](#)
  - nas\_lteWcdmaCellInfo, [431](#)
- chaddr
  - WdsDHCPv4HWConfig, [1057](#)
  - wdsDhcpv4HwConfig, [1056](#)
- chaddrLen
  - WdsDHCPv4HWConfig, [1057](#)
  - wdsDhcpv4HwConfig, [1056](#)
- changePIN
  - pack\_uim\_ChangePin\_t, [560](#)
  - UIMChangePinReq, [812](#)
- changeUIMPIN, [155](#)
  - oldPINLen, [156](#)
  - oldPINVal, [156](#)
  - pinID, [156](#)
  - pinLen, [156](#)
  - pinVal, [156](#)
- ChannelRate, [156](#)
  - CurrChanRxRate, [156](#)
  - CurrChanTxRate, [156](#)
  - DUNCallInfoInd, [209](#)
  - MaxChanRxRate, [156](#)
  - MaxChanTxRate, [156](#)
- channelRate, [157](#)
  - CurrChanRxRate, [157](#)
  - CurrChanTxRate, [157](#)
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [964](#)
- Chipset
  - DeviceConfigDetail, [197](#)
- ckLen
  - depersonalizationInformation, [195](#)
- ckVal
  - depersonalizationInformation, [195](#)
- ClkInfo, [159](#)

- mask, [160](#)
- codingScheme
  - PLMNNetworkNameData, [596](#)
  - remotePartyName, [658](#)
- CommInfo, [162](#)
  - imsRegState, [163](#)
  - modemMode, [163](#)
  - psState, [163](#)
  - systemMode, [163](#)
  - temperature, [163](#)
- common.h
  - eCTL, [1071](#)
  - eDMS, [1071](#)
  - eIND, [1072](#)
  - eLOC, [1071](#)
  - eLOG\_DEBUG, [1071](#)
  - eLOG\_FATAL, [1071](#)
  - eLOG\_INFO, [1071](#)
  - eLOG\_WARN, [1071](#)
  - eNAS, [1071](#)
  - eQOS, [1071](#)
  - eREQ, [1072](#)
  - eRSP, [1072](#)
  - eSMS, [1071](#)
  - eSWILOC, [1072](#)
  - eSWIOMA, [1072](#)
  - eTIMEOUT\_10\_S, [1072](#)
  - eTIMEOUT\_20\_S, [1072](#)
  - eTIMEOUT\_2\_S, [1072](#)
  - eTIMEOUT\_300\_S, [1072](#)
  - eTIMEOUT\_30\_S, [1072](#)
  - eTIMEOUT\_5\_S, [1072](#)
  - eTIMEOUT\_60\_S, [1072](#)
  - eTIMEOUT\_8\_S, [1072](#)
  - eTIMEOUT\_DEFAULT, [1072](#)
  - eTMD, [1072](#)
  - eUIM, [1071](#)
  - eWDS, [1071](#)
- common.h, [1069](#)
  - eLOG\_LEVEL, [1071](#)
  - eQMI\_SVC, [1071](#)
  - eTimeout, [1072](#)
  - fill\_pack\_ctx, [1072](#)
  - fill\_sdu\_hdr, [1072](#)
  - get\_version, [1072](#)
  - glog, [1073](#)
  - gloglvl, [1073](#)
  - helper\_get\_resp\_ctx, [1072](#)
  - helper\_get\_xid, [1073](#)
  - helper\_set\_log\_func, [1073](#)
  - helper\_set\_log\_lvl, [1073](#)
  - libpack\_GetVersion, [1073](#)
  - libpack\_log, [1073](#)
  - logger, [1071](#)
  - MINREQBKLEN, [1071](#)
  - MSGID\_AND\_LEN, [1071](#)
  - MSGID\_DONT\_CARE, [1071](#)
  - msgtype, [1072](#)
  - SDU\_HDR\_LEN, [1071](#)
  - UNUSEDPARAM, [1071](#)
  - unpack\_result\_code\_only, [1073](#)
- commonInfo
  - swiModemStatusResp, [760](#)
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, [909](#)
- ConcSvcInfo
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- concSvcInfo
  - qaQmiServingSystemParam, [619](#)
- conn\_status
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [968](#)
- ConnRateElmntsV4
  - WdsConnectionRate, [1053](#)
- ConnRateElmntsV6
  - WdsConnectionRate, [1053](#)
- connStatus
  - \_packetSrvStatus, [62](#)
- connectNumInfo, [164](#)
  - callerID, [166](#)
  - callerIDLen, [166](#)
  - numPlan, [166](#)
  - numPresInd, [166](#)
  - numType, [166](#)
  - screeningInd, [166](#)
- ConnectedPartyNum
  - arrConnectPartyNum, [111](#)
- ConnectionStatus, [163](#)
  - MDMCallDuration, [164](#)
  - MDMConnStatus, [164](#)
- connectionStatus, [164](#)
  - MDMCallDuration, [164](#)
  - MDMConnStatus, [164](#)
  - unpack\_wds\_GetSessionState\_t, [961](#)
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [964](#)
- connetionState
  - imsaPdpStatusInfo, [288](#)
- content
  - authenticateResult, [115](#)
  - readResult, [654](#)
  - uim\_readResult, [803](#)
- contentLen
  - authenticateResult, [115](#)
  - readResult, [654](#)
  - uim\_readResult, [803](#)
- context
  - authenticationData, [115](#)
- contextId
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [579](#)
  - swiPDPRuntimeSettingsReq, [762](#)
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [972](#)
- contextType
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [579](#)
  - swiPDPRuntimeSettingsReq, [762](#)

- ControlMac
  - protocolSubtypeElement, [613](#)
- Count1
  - RankIndicatorInd, [653](#)
- Count2
  - RankIndicatorInd, [653](#)
- countryInitials
  - PLMNNetworkNameData, [596](#)
- coverage
  - altitudeSrcInfo, [100](#)
- cpich\_ecno
  - nas\_wcdmaCellInfo, [462](#)
  - wcdmaCellInfo, [1031](#)
- cpich\_rscp
  - nas\_wcdmaCellInfo, [462](#)
  - wcdmaCellInfo, [1031](#)
- cradleMountConfigStatus
  - QmiCbkLocCradleMountInd, [626](#)
- crashAction
  - pack\_dms\_SetCrashAction\_t, [521](#)
- crashData
  - CrashInfo, [167](#)
- crashId
  - CrashInfo, [167](#)
- CrashInfo, [166](#)
  - crashData, [167](#)
  - crashId, [167](#)
  - crashStrLen, [167](#)
  - gcDumpStrLen, [167](#)
  - numCrashes, [167](#)
  - pCrashString, [167](#)
  - pGCDumpString, [167](#)
- CrashInfoParams, [167](#)
  - pCrashInfo, [168](#)
  - pDevCrashStatus, [168](#)
- crashStrLen
  - CrashInfo, [167](#)
- CreateProfileIn, [168](#)
  - curProfile, [168](#)
  - pProfileID, [169](#)
  - pProfileType, [169](#)
- CreateProfileOut, [169](#)
  - pExtErrorCode, [169](#)
  - pProfileIndex, [169](#)
  - pProfileType, [169](#)
- csAttachState
  - nas\_servSystem, [446](#)
  - NASServingSystemInfo, [494](#)
  - ServingSystemInfo, [680](#)
  - servSystem, [681](#)
- csBarStatus
  - CallBarringSysInfo, [122](#)
  - callBarStatus, [123](#)
  - nas\_CallBarringSysInfo, [387](#)
  - nas\_callBarStatus, [388](#)
- cur\_carr\_name
  - slqsfwinfo\_s, [717](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [849](#)
- cur\_carr\_rev
  - slqsfwinfo\_s, [717](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [849](#)
- curAMRConfig, [171](#)
  - gsmAmrStat, [171](#)
  - wcdmaAmrStat, [172](#)
- curDataBearerTechnology
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, [963](#)
- curProfile
  - \_GetProfileSettingOut, [57](#)
  - CreateProfileIn, [168](#)
  - ModifyProfileIn, [383](#)
  - pack\_wds\_SLQSModifyProfile\_t, [575](#)
  - UnPackGetProfileSettingOut, [973](#)
- CurrChanRxRate
  - ChannelRate, [156](#)
  - channelRate, [157](#)
  - dunchannelRate, [210](#)
- CurrChanTxRate
  - ChannelRate, [156](#)
  - channelRate, [157](#)
  - dunchannelRate, [210](#)
- currDBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- CurrDataSysStat, [172](#)
  - pCurrNetworkInfo, [172](#)
  - pNetworkInfoLen, [172](#)
  - pPrefNetwork, [172](#)
- CurrImageInfo, [175](#)
  - buildID, [176](#)
  - buildIDLen, [176](#)
  - imageType, [176](#)
  - uniqueID, [176](#)
- currNWInfo
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- CurrNetworkInfo, [176](#)
  - NetworkType, [178](#)
  - RATMask, [178](#)
  - SOMask, [178](#)
- currNetworkInfo, [178](#)
  - NetworkType, [178](#)
  - RATMask, [179](#)
  - SOMask, [179](#)
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, [963](#)
- current\_channel\_rx\_rate
  - WDSSWICurrentChannelRates, [1068](#)
- current\_channel\_tx\_rate
  - WDSSWICurrentChannelRates, [1068](#)
- currentCatEvent, [172](#)
  - CatAlphaIdtfr, [173](#)
  - CatEndPS, [173](#)
  - CatEvIDData, [173](#)
  - CatEventLst, [173](#)
  - CatRefresh, [173](#)

- currentChannelRXRate
  - unpack\_wds\_GetConnectionRate\_t, 955
- currentChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, 955
- currentDataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 577
- CurrentImgList, 173
  - carrier, 174
  - fwvers, 174
  - numEntries, 174
  - pCurrImgInfo, 174
  - pkgver, 174
  - priver, 174
- currentMitigationLvl
  - QmiCbkTmdMitiLvlRptInd, 639
- currentNetwork
  - dataBearerTechnology, 189
  - qmiWSDDataBearerTechnology, 643
- CurrentPLMN
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- currentPLMN, 174
  - MCC, 175
  - MNC, 175
  - netDescr, 175
  - netDescrLength, 175
- cust\_attr
  - custSettingInfo, 183
  - DMScustSettingInfo, 202
- cust\_id
  - custSettingInfo, 183
  - DMScustSettingInfo, 202
  - DMSgetCustomInput, 204
  - getCustomInput, 238
  - pack\_dms\_GetCustFeaturesV2\_t, 521
  - pack\_dms\_SetCustFeaturesV2\_t, 522
  - setCustomSettingV2, 688
- cust\_value
  - custSettingInfo, 183
  - DMScustSettingInfo, 202
  - pack\_dms\_SetCustFeaturesV2\_t, 522
  - setCustomSettingV2, 688
- custFeaturesInfo, 179
  - GpsEnable, 181
  - pDHCPRelayEnabled, 181
  - pDisableIMSI, 181
  - pGPSLPM, 181
  - pGPSSel, 181
  - pIPFamSupport, 181
  - plsVoiceEnabled, 181
  - pRMAutoConnect, 181
  - pSMSSupport, 181
  - qaGobiApiDms.h, 1248
- custFeaturesSetting, 181
  - pDHCPRelayEnabled, 183
  - pGPSEnable, 183
  - pGPSLPM, 183
  - pGPSSel, 183
  - plsVoiceEnabled, 183
  - qaGobiApiDms.h, 1249
- custSetting
  - custSettingList, 184
  - DMScustSettingList, 203
- custSettingInfo, 183
  - cust\_attr, 183
  - cust\_id, 183
  - cust\_value, 183
  - id\_length, 184
  - value\_length, 184
- custSettingList, 184
  - custSetting, 184
  - list\_type, 184
  - num\_instances, 184
- CustomSCP
  - unpack\_nas\_GetCDMANetworkParameters\_t, 885
- CwtMute
  - GetM2MAudioProfileResp, 251
  - GetM2MAVMuteResp, 253
- DEVICE\_STATE\_BOOT
  - qaGobiApiCbk.h, 1200
- DEVICE\_STATE\_DISCONNECTED
  - qaGobiApiCbk.h, 1200
- DEVICE\_STATE\_READY
  - qaGobiApiCbk.h, 1200
- dBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- dBTechnology
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- DEFAULTBYTEVALUE
  - qaGobiApiPds.h, 1367
- DEFAULTLONGVALUE
  - qaGobiApiPds.h, 1367
- DEFAULTWORDVALUE
  - qaGobiApiPds.h, 1367
- DEREGISTER\_EVENT
  - qaGobiApiCbk.h, 1164
- DEREGISTER\_SRV
  - qaGobiApiCbk.h, 1164
- DEVICE\_OFFLINE
  - qaGobiApiFms.h, 1286
- DEVICE\_RESET
  - qaGobiApiFms.h, 1286
- DEVICE\_SHUTDOWN
  - qaGobiApiFms.h, 1286
- DHCPOption, 198
  - optCode, 198
  - optValLen, 198
  - pOptVal, 198
- DHCPOptionList, 198
  - numOpt, 199
  - pOptions, 199
- DHCPRelayEnabled
  - pack\_dms\_SetCustFeature\_t, 522
  - unpack\_dms\_GetCustFeature\_t, 845

- DMS\_IMGDETAILS\_LEN
  - dms.h, [1077](#)
- DMS\_PM\_FACTORY
  - dms.h, [1078](#)
- DMS\_PM\_LOW
  - dms.h, [1078](#)
- DMS\_PM\_OFFLINE
  - dms.h, [1078](#)
- DMS\_PM\_ONLINE
  - dms.h, [1078](#)
- DMS\_PM\_RESET
  - dms.h, [1078](#)
- DMS\_PM\_SHUT\_DOWN
  - dms.h, [1078](#)
- DMScustSettingInfo, [202](#)
  - cust\_attr, [202](#)
  - cust\_id, [202](#)
  - cust\_value, [202](#)
  - id\_length, [202](#)
  - value\_length, [202](#)
- DMScustSettingList, [203](#)
  - custSetting, [203](#)
  - list\_type, [203](#)
  - num\_instances, [203](#)
- DMSgetCustomFeatureV2, [203](#)
  - pCustSettingInfo, [204](#)
  - pCustSettingList, [204](#)
  - pGetCustomInput, [204](#)
- DMSgetCustomInput, [204](#)
  - cust\_id, [204](#)
  - list\_type, [204](#)
- DRCover
  - DRCParams, [207](#)
- DRCParams, [207](#)
  - DRCover, [207](#)
  - DRCValue, [207](#)
- DRCValue
  - DRCParams, [207](#)
- DTMFEvent
  - DTMFInfo, [208](#)
- DTMFInfo, [207](#)
  - callID, [208](#)
  - DTMFEvent, [208](#)
  - digitBuff, [208](#)
  - digitCnt, [208](#)
- DTMFInformation
  - voiceDTMFEventInfo, [990](#)
- DTMFInterdigitInterval
  - DTMFLengths, [209](#)
- DTMFLengths, [208](#)
  - DTMFInterdigitInterval, [209](#)
  - DTMFPulseWidth, [209](#)
- DTMFPulseWidth
  - DTMFLengths, [209](#)
- DTMFdigit
  - voiceContDTMFInfo, [989](#)
- DTMInd
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetservingSystem\_t, [897](#)
- DUNCallInfoInd, [209](#)
  - CallEndReason, [209](#)
  - ChannelRate, [209](#)
  - DataBearerTech, [209](#)
  - DormancyStatus, [209](#)
  - MdmConnStatus, [209](#)
  - RXOKBytesCount, [209](#)
  - TXOKBytesCount, [209](#)
- Data
  - CatEventIDDDataTlv, [138](#)
- data
  - authenticationData, [116](#)
  - SMSCAddress, [733](#)
  - sMSCAddress, [733](#)
  - SMSEtwsMessage, [735](#)
  - sMSEtwsMessage, [734](#)
  - SMSTransferRouteMTMessage, [746](#)
  - sMSTransferRouteMTMessage, [745](#)
  - SwiOTAMsg\_s, [761](#)
- data\_buf
  - NASOTAMessageTlv, [484](#)
- data\_len
  - NASOTAMessageTlv, [484](#)
  - SwiOTAMsg\_s, [761](#)
- dataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [577](#)
- dataBearerMask
  - dataBearers, [185](#)
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, [963](#)
- DataBearerTech, [185](#)
  - DUNCallInfoInd, [209](#)
  - ratValue, [187](#)
  - soMask, [187](#)
  - techType, [187](#)
- dataBearerTech
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [964](#)
- DataBearerTechExt, [187](#)
  - pBearerTech, [188](#)
  - pLastBearerTech, [188](#)
- dataBearerTechnology, [188](#)
  - currentNetwork, [189](#)
  - ratMask, [189](#)
  - soMask, [189](#)
- dataBearers, [184](#)
  - dataBearerMask, [185](#)
  - pCurDataBearerTechnology, [185](#)
  - pLastCallDataBearerTechnology, [185](#)
- dataCapabilities
  - dataSrvCapabilities, [190](#)
  - nas\_dataSrvCapabilities, [399](#)
- dataCapabilitiesLen
  - dataSrvCapabilities, [190](#)
  - nas\_dataSrvCapabilities, [399](#)
- DataCaps
  - unpack\_nas\_GetServingNetwork\_t, [888](#)

- unpack\_nas\_GetServingNetworkCapabilities\_t, 889
- dataCaps
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 890
- DataCapsLen
  - unpack\_nas\_GetServingNetwork\_t, 888
  - unpack\_nas\_GetServingNetworkCapabilities\_t, 889
- dataCapsSize
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 890
- dataLen
  - authenticationData, 116
- DataLength
  - CatEventIDDDataTlv, 138
- DataRate
  - unpack\_qos\_swiQosFlow\_t, 935
- dataRate, 189
  - dataRateMax, 190
  - guaranteedRate, 190
- dataRateMax
  - dataRate, 190
  - unpack\_qos\_dataRate\_t, 918
- dataServiceCaCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, 847
- DataServiceCapability
  - unpack\_dms\_GetDeviceCap\_t, 846
- DataSrvCapabilities
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- dataSrvCapabilities, 190
  - dataCapabilities, 190
  - dataCapabilitiesLen, 190
- DataStatusDetail, 191
  - IPAddress, 192
  - LastErrCode, 192
- dataSysStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- dataSystemStatus
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 577
- DataULongLongTlv, 193
  - TlvPresent, 193
  - ulldata, 193
- DataULongTlv, 193
  - TlvPresent, 193
  - ulldata, 193
- Date
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 946
  - wcdmaLongMsgDecodingParams, 1034
  - wcdmaMsgDecodingParams, 1036
- DateLength
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 946
- day
  - nas\_timeInfo, 456
- nas\_UniversalTime, 461
  - timeInfo, 784
  - UniversalTime, 843
- dayLtSavingAdj
  - nas\_timeInfo, 456
  - timeInfo, 784
- dayOfWeek
  - nas\_timeInfo, 456
  - nas\_UniversalTime, 461
  - timeInfo, 784
  - UniversalTime, 843
- daylightSavings
  - nas\_qaQmi3Gpp2TimeZone, 439
  - qaQmi3Gpp2TimeZone, 615
- DcsUsbPortNames, 193
  - AtCmdPort, 193
  - DmPort, 193
  - NmeaPort, 193
- defaultPDNEnabled
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 576
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 962
- DefaultRoamInd
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- defaultRoamInd
  - qaQmiServingSystemParam, 619
- delAssistDataStatus, 193
  - status, 194
- delayClass
  - GPRSQoS, 261
  - GPRSRequestedQoS, 262
  - LibPackGPRSRequestedQoS, 307
  - wds\_GPRSQoS, 1045
- DeleteStoredImage
  - qaGobiApiFms.h, 1289
- deliveryErrSDU
  - LibPackUMTSQoS, 334
  - UMTSMinQoS, 837
  - UMTSQoS, 840
  - wds\_UMTSMinQoS, 1050
- depersonalizationInformation, 194
  - ckLen, 195
  - ckVal, 195
  - feature, 195
  - operation, 195
- depersonalisationInfo
  - UIMDepersonalizationReq, 813
- Description
  - SlqsNas3GppNetworkInfo, 718
- description
  - omaDmFotaTlv, 515
  - omaDmFotaTlvExt, 518
  - unpack\_omaDmFotaTlv\_t, 917
- descriptionlength
  - omaDmFotaTlv, 515
  - omaDmFotaTlvExt, 518
  - unpack\_omaDmFotaTlv\_t, 917
- Desription
  - nas\_QmiNas3GppNetworkInfo, 440

- destPortRangeEnd
  - LibPackTFTIDParams, 331
  - TFTIDParams, 782
- destPortRangeStart
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- detailSvcInfo, 195
  - hdrHybrid, 196
  - hdrSrvStatus, 196
  - isSysForbidden, 196
  - srvCapability, 196
  - srvStatus, 197
- DetailedSvcInfo
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- dev
  - qmifwinfo\_s, 640
- DevCrashState
  - unpack\_dms\_GetCrashAction\_t, 844
- Device
  - SetM2MAudioAVCFGRReq, 694
- Device Connectivity Service (DCS), 31
- Device Management Service (DMS), 33
- device\_state\_enum
  - qaGobiApiCbK.h, 1200
- DeviceConfigDetail, 197
  - Chipset, 197
  - HWVersion, 197
  - QLIC, 198
  - Technology, 198
- deviceId
  - \_MitigationDevInfo, 59
- deviceIdLen
  - \_MitigationDevInfo, 59
- DiagInfo
  - allCallsDiagInfo, 98
  - arrDiagInfo, 111
- diagInfo, 199
  - diagInfoLen, 199
  - diagnosticInfo, 199
- diagInfoLen
  - diagInfo, 199
- diagnosticInfo
  - diagInfo, 199
- digitBuff
  - DTMFInfo, 208
- digitCnt
  - burstDTMFInfo, 121
  - DTMFInfo, 208
- dirNum, 199
  - dirNum, 200
  - dirNumLen, 200
  - dirNum, 200
- dirNumLen
  - dirNum, 200
- direction
  - callInfo, 132
- DisableIMSI
  - pack\_dms\_SetCustFeature\_t, 522
  - unpack\_dms\_GetCustFeature\_t, 845
- dispType
  - extDispRecInfo, 215
- displayCondition
  - serviceName, 678
- dl\_bw\_value
  - nas\_PhyCaAggPcellInfo, 434
  - nas\_PhyCaAggScellIDBw, 435
  - nas\_PhyCaAggScellInfo, 439
  - NASPhyCaAggPcellInfo, 485
  - NASPhyCaAggScellIDBw, 486
  - NASPhyCaAggScellInfo, 488
  - PhyCaAggPcellInfo, 589
  - PhyCaAggScellIDBw, 589
  - PhyCaAggScellInfo, 593
- DmPort
  - DcsUsbPortNames, 193
- dms.h, 1073
  - DMS\_IMGDETAILS\_LEN, 1077
  - DMS\_PM\_FACTORY, 1078
  - DMS\_PM\_LOW, 1078
  - DMS\_PM\_OFFLINE, 1078
  - DMS\_PM\_ONLINE, 1078
  - DMS\_PM\_RESET, 1078
  - DMS\_PM\_SHUT\_DOWN, 1078
  - MAX\_BUILD\_ID\_LEN, 1078
  - pack\_dms\_GetActivationState, 1079
  - pack\_dms\_GetBandCapability, 1079
  - pack\_dms\_GetCrashAction, 1079
  - pack\_dms\_GetCustFeature, 1080
  - pack\_dms\_GetCustFeaturesV2, 1080
  - pack\_dms\_GetDeviceCap, 1080
  - pack\_dms\_GetDeviceCapabilities, 1080
  - pack\_dms\_GetDeviceHardwareRev, 1081
  - pack\_dms\_GetDeviceMfr, 1081
  - pack\_dms\_GetDeviceSerialNumbers, 1081
  - pack\_dms\_GetFSN, 1083
  - pack\_dms\_GetFirmwareInfo, 1082
  - pack\_dms\_GetFirmwareRevision, 1082
  - pack\_dms\_GetFirmwareRevisions, 1082
  - pack\_dms\_GetHardwareRevision, 1083
  - pack\_dms\_GetIMSI, 1083
  - pack\_dms\_GetModelID, 1084
  - pack\_dms\_GetNetworkTime, 1084
  - pack\_dms\_GetPRLVersion, 1085
  - pack\_dms\_GetPower, 1084
  - pack\_dms\_GetSerialNumbers, 1085
  - pack\_dms\_GetUSBComp, 1086
  - pack\_dms\_GetVoiceNumber, 1086
  - pack\_dms\_SLQSDmsSwiGetResetInfo, 1089
  - pack\_dms\_SLQSDmsSwiIndicationRegister, 1089
  - pack\_dms\_SLQSGetBandCapability, 1090
  - pack\_dms\_SLQSSwiClearDyingGaspStatistics, 1090
  - pack\_dms\_SLQSSwiGetDyingGaspCfg, 1091
  - pack\_dms\_SLQSSwiGetDyingGaspStatistics, 1091

- pack\_dms\_SLQSSwiGetFirmwareCurr, [1091](#)
- pack\_dms\_SLQSSwiGetFwUpdateStatus, [1092](#)
- pack\_dms\_SLQSSwiSetDyingGaspCfg, [1092](#)
- pack\_dms\_SetCrashAction, [1086](#)
- pack\_dms\_SetCustFeature, [1087](#)
- pack\_dms\_SetCustFeaturesV2, [1087](#)
- pack\_dms\_SetEventReport, [1088](#)
- pack\_dms\_SetFirmwarePreference, [1088](#)
- pack\_dms\_SetPower, [1088](#)
- pack\_dms\_SetUSBComp, [1089](#)
- pack\_dms\_UIMGetICCID, [1092](#)
- UNIQUE\_ID\_LEN, [1078](#)
- unpack\_dms\_GetActivationState, [1093](#)
- unpack\_dms\_GetBandCapability, [1093](#)
- unpack\_dms\_GetCrashAction, [1093](#)
- unpack\_dms\_GetCustFeature, [1094](#)
- unpack\_dms\_GetCustFeaturesV2, [1094](#)
- unpack\_dms\_GetDeviceCap, [1094](#)
- unpack\_dms\_GetDeviceCapabilities, [1095](#)
- unpack\_dms\_GetDeviceHardwareRev, [1095](#)
- unpack\_dms\_GetDeviceMfr, [1095](#)
- unpack\_dms\_GetDeviceSerialNumbers, [1096](#)
- unpack\_dms\_GetFSN, [1097](#)
- unpack\_dms\_GetFirmwareInfo, [1096](#)
- unpack\_dms\_GetFirmwareRevision, [1096](#)
- unpack\_dms\_GetFirmwareRevisions, [1097](#)
- unpack\_dms\_GetHardwareRevision, [1097](#)
- unpack\_dms\_GetIMSI, [1098](#)
- unpack\_dms\_GetModelID, [1098](#)
- unpack\_dms\_GetNetworkTime, [1098](#)
- unpack\_dms\_GetPRLVersion, [1099](#)
- unpack\_dms\_GetPower, [1099](#)
- unpack\_dms\_GetSerialNumbers, [1099](#)
- unpack\_dms\_GetUSBComp, [1100](#)
- unpack\_dms\_GetVoiceNumber, [1100](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo, [1103](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind, [1103](#)
- unpack\_dms\_SLQSDmsSwiIndicationRegister, [1104](#)
- unpack\_dms\_SLQSGetBandCapability, [1104](#)
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics, [1105](#)
- unpack\_dms\_SLQSSwiGetDyingGaspCfg, [1105](#)
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics, [1105](#)
- unpack\_dms\_SLQSSwiGetFirmwareCurr, [1106](#)
- unpack\_dms\_SLQSSwiGetFwUpdateStatus, [1106](#)
- unpack\_dms\_SLQSSwiSetDyingGaspCfg, [1106](#)
- unpack\_dms\_SetCrashAction, [1100](#)
- unpack\_dms\_SetCustFeature, [1101](#)
- unpack\_dms\_SetCustFeaturesV2, [1101](#)
- unpack\_dms\_SetEventReport, [1101](#)
- unpack\_dms\_SetEventReport\_ind, [1102](#)
- unpack\_dms\_SetFirmwarePreference, [1102](#)
- unpack\_dms\_SetPower, [1102](#)
- unpack\_dms\_SetUSBComp, [1103](#)
- unpack\_dms\_UIMGetICCID, [1107](#)
- dms\_ActivationStatusTlv, [200](#)
  - activationStatus, [200](#)
  - TlvPresent, [200](#)
- dms\_OperatingModeTlv, [201](#)
  - operatingMode, [201](#)
  - TlvPresent, [201](#)
- dmsCurrentPRLInfo, [201](#)
  - pPRLPreference, [202](#)
  - pPRLVersion, [202](#)
  - qaGobiApiDms.h, [1250](#)
- dmsIndicationRegisterReq, [204](#)
  - pSwiGetResetInd, [205](#)
- dmsSwiGetResetInfo, [205](#)
  - source, [205](#)
  - type, [206](#)
- Domain, [206](#)
  - domainLen, [206](#)
  - domainName, [206](#)
- domain
  - DomainNameList, [206](#)
  - wds\_DomainNameList, [1044](#)
- domainLen
  - Domain, [206](#)
  - wds\_Domain, [1044](#)
- DomainList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [966](#)
- domainName
  - Domain, [206](#)
  - wds\_Domain, [1044](#)
- DomainNameList, [206](#)
  - domain, [206](#)
  - numInstances, [207](#)
- dormancyStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- dormancyState
  - unpack\_wds\_GetDormancyState\_t, [957](#)
- DormancyStatus
  - DUNCallInfoInd, [209](#)
- dormancyStatus
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [577](#)
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [964](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- downLink
  - NSSAudioCtrl, [511](#)
- dscp
  - QosMap, [653](#)
- dtmSupp
  - GSMSysInfo, [271](#)
  - nas\_GSMSysInfo, [410](#)
- dtmSuppValid
  - GSMSysInfo, [271](#)
  - nas\_GSMSysInfo, [410](#)
- dunchannelRate, [210](#)
  - CurrChanRxRate, [210](#)
  - CurrChanTxRate, [210](#)
  - MaxChanRxRate, [210](#)

- MaxChanTxRate, [210](#)
- Duration
  - pack\_nas\_SetNetworkPreference\_t, [533](#)
  - unpack\_nas\_GetNetworkPreference\_t, [887](#)
- eCTL
  - common.h, [1071](#)
- eDMS
  - common.h, [1071](#)
- eGOBI\_DEV\_SERIES\_9X15
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_DEV\_SERIES\_9X30
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_DEV\_SERIES\_G3K
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_DEV\_SERIES\_NON\_GOBI
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_DEV\_SERIES\_SIERRA\_GOBI
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_DEV\_SERIES\_UNKNOWN
  - qaGobiApiFms.h, [1287](#)
- eGOBI\_IMG\_CAR\_3
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_AERIS
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_ALLTEL
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_AMX\_TELCEL
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_ATT
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_BELL
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_BHARTI
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_BRASIL\_VIVO
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_CHINA\_MOBILE
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_CHINA\_TELECOM
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_CHINA\_UNICOM
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_EMOBILE
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_FACTORY
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_GENERIC
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_GENERIC\_CDMA
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_IUSACELL
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_KDDI
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_KT\_FREETEL
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_LEAP
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_METROPCS
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_NETCOM
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_NORF
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_NTT\_DOCOMO
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_O2
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_OMH
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_ORANGE
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_RELIANCE1
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_RELIANCE2
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_ROGERS
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_SFR
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SINGTEL\_OPTUS
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SK\_TELCOM1
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SK\_TELCOM2
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SOFTBANK
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SPRINT
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_SWISSCOM
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TATA
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELCOM\_ITALIA
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELCOM\_NZ
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELEFONICA
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELNOR
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_TELIASONERA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_CAR\_TELSTRA1
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELSTRA2
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TELUS
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_TMOBILE
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_US
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_CAR\_VERIZON
  - qaGobiApiFms.h, [1288](#)

- eGOBI\_IMG\_CAR\_VODAFONE
  - qaGobiApiFms.h, [1288](#)
- eGOBI\_IMG\_GPS\_ASSISTED
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_GPS\_NO\_XTRA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_GPS\_NONE
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_GPS\_STAND\_ALONE
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_ASIA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_AUS
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_EU
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_GLOBAL
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_LA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_REG\_NA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_TECH\_CDMA
  - qaGobiApiFms.h, [1289](#)
- eGOBI\_IMG\_TECH\_UMTS
  - qaGobiApiFms.h, [1289](#)
- eGobi\_DEV\_SERIES\_MC83
  - qaGobiApiFms.h, [1287](#)
- eIND
  - common.h, [1072](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED
  - nas.h, [1127](#)
- eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED
  - nas.h, [1127](#)
- eLOC
  - common.h, [1071](#)
- eLOG\_DEBUG
  - common.h, [1071](#)
- eLOG\_FATAL
  - common.h, [1071](#)
- eLOG\_INFO
  - common.h, [1071](#)
- eLOG\_WARN
  - common.h, [1071](#)
- eNAS
  - common.h, [1071](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_100
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_15
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_25
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_50
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_6
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_75
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED\_LITE
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED\_LITE
  - nas.h, [1127](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED
  - qaGobiApiNas.h, [1334](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED\_LITE
  - nas.h, [1127](#)
- eNAS\_RADIO\_IF\_GSM
  - qaGobiApiNas.h, [1333](#)
- eNAS\_RADIO\_IF\_LTE
  - qaGobiApiNas.h, [1333](#)
- eNAS\_RADIO\_IF\_TDSCDMA
  - qaGobiApiNas.h, [1333](#)
- eNAS\_RADIO\_IF\_UMTS
  - qaGobiApiNas.h, [1333](#)
- eQA\_QMI\_SVC\_NA
  - qaGobiApiCbK.h, [1200](#)
- eQA\_QMI\_SVC\_NAS
  - qaGobiApiCbK.h, [1200](#)

eQA\_QMI\_SVC\_WDS  
     qaGobiApiCbk.h, 1200  
 eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_BUFFER\_SZ  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_CANCEL\_OP  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_DRIVER  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_ENUM\_BEGIN  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_ENUM\_END  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_FILE\_COPY  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_FILE\_OPEN  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_GENERAL  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INTERNAL  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INVALID\_ARG  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INVALID\_DEVID  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INVALID\_FILE  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INVALID\_QMI\_RSP  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_INVALID\_XID  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_MEMORY  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_MULTIPLE\_DEVICES  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_NO\_CANCELABLE\_OP  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_NO\_CONNECTION  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_NO\_DEVICE  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_NO\_SIGNAL  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_NONE  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_NULL\_TLV  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_OFFLINE  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_PDU\_GENERATION  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_ABORTED  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_CALL\_FAILED  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_CAT\_END  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_QMI\_CAT\_START  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_QMI\_CAUSE\_CODE  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_CONNECT  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE  
     qmerrno.h, 1549  
 eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_DISABLED  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_ENCODING  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED  
     qmerrno.h, 1552  
 eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_GENERAL  
     qmerrno.h, 1550  
 eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED  
     qmerrno.h, 1551  
 eQCWWAN\_ERR\_QMI\_IFACE  
     qmerrno.h, 1548  
 eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE  
     qmerrno.h, 1551

- eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCE-  
S  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INTERNAL  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ARG  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD  
qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ID  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_INDEX  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTI-  
ON  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTIO-  
N  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP  
qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_MAX  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_I-  
N\_USE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_-  
USE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAIL-  
URE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE  
qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED  
qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_OFFSET  
qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTE-  
D

- qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED
  - qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL
  - qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_REQ
  - qmerrno.h, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH
  - qmerrno.h, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO
  - qmerrno.h, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUPPORTED
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_RSP
  - qmerrno.h, [1548](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO
  - qmerrno.h, [1548](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION
  - qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN
  - qmerrno.h, [1550](#)
- eQCWWAN\_ERR\_QMI\_WIDTH
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_RESET
  - qmerrno.h, [1549](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_END
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_START
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP
  - qmerrno.h, [1551](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_END
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_START
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_END
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND

- qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWN-  
LOADED
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOAD-  
\_MODE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISM-  
ATCH
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT-  
\_ACTIVE\_IMAGE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_START
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISM\_END
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_F-  
OUND
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPT-  
ED
  - qmerrno.h, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_START
  - qmerrno.h, [1552](#)
- eQMI\_LOC\_SESS\_STATUS\_FAILURE
  - loc.h, [1115](#)
- eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS
  - loc.h, [1115](#)
- eQMI\_LOC\_SESS\_STATUS\_SUCCESS
  - loc.h, [1115](#)
- eQMI\_LOC\_SESS\_STATUS\_TIMEOUT
  - loc.h, [1115](#)
- eQOS
  - common.h, [1071](#)
- eREQ
  - common.h, [1072](#)
- eRSP
  - common.h, [1072](#)
- eSMS
  - common.h, [1071](#)
- eSWILOC
  - common.h, [1072](#)
- eSWIOMA
  - common.h, [1072](#)
- eSYS\_SRV\_DOMAIN\_CAMPED
  - qaGobiApiNas.h, [1333](#)
- eSYS\_SRV\_DOMAIN\_CS\_ONLY
  - qaGobiApiNas.h, [1333](#)
- eSYS\_SRV\_DOMAIN\_CS\_PS
  - qaGobiApiNas.h, [1333](#)
- eSYS\_SRV\_DOMAIN\_NO\_SRV
  - qaGobiApiNas.h, [1333](#)
- eSYS\_SRV\_DOMAIN\_PS\_ONLY
  - qaGobiApiNas.h, [1333](#)
- eSYS\_SRV\_DOMAIN\_UNKNOWN
  - qaGobiApiNas.h, [1333](#)
- eSetServiceAutomaticTrackingDisable
  - qaGobiApiPds.h, [1367](#)
- eSetServiceAutomaticTrackingEnable
  - qaGobiApiPds.h, [1367](#)
- eTIMEOUT\_10\_S
  - common.h, [1072](#)
- eTIMEOUT\_20\_S
  - common.h, [1072](#)
- eTIMEOUT\_2\_S
  - common.h, [1072](#)
- eTIMEOUT\_300\_S
  - common.h, [1072](#)
- eTIMEOUT\_30\_S
  - common.h, [1072](#)
- eTIMEOUT\_5\_S
  - common.h, [1072](#)
- eTIMEOUT\_60\_S
  - common.h, [1072](#)
- eTIMEOUT\_8\_S
  - common.h, [1072](#)
- eTIMEOUT\_DEFAULT
  - common.h, [1072](#)
- eTLV\_3GPP\_NETWORK\_INFO
  - qaNasPerformNetworkScan.h, [1546](#)
- eTLV\_CBK\_ALPHA\_IDENTIFIER
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_DISPLAY\_TEXT
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_END\_PROACTIVE\_SESSION
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_GET\_IN\_KEY
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_GET\_INPUT
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_LANGUAGE\_NOTIFICATION
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_REFRESH
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_SELECT\_ITEM

- qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_SETUP\_EVENT\_LIST
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_SETUP\_IDLE\_MODE\_TEXT
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_CBK\_SETUP\_MENU
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_END\_PROACTIVE\_SESSION\_LENGTH
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_IND\_OMA\_DM\_CONFIG
  - qaCbkSwiOmaDmEventReportInd.h, [1151](#)
- eTLV\_IND\_OMA\_DM\_FOTA
  - qaCbkSwiOmaDmEventReportInd.h, [1151](#)
- eTLV\_IND\_OMA\_DM\_NOT
  - qaCbkSwiOmaDmEventReportInd.h, [1151](#)
- eTLV\_REFRESH\_LENGTH
  - qaCbkCatEventReportInd.h, [1150](#)
- eTLV\_RF\_BAND\_INFO
  - qaNasGetRFBandInfo.h, [1545](#)
- eTLV\_SETUP\_EVENT\_LIST\_LENGTH
  - qaCbkCatEventReportInd.h, [1150](#)
- eTMD
  - common.h, [1072](#)
- eUIM
  - common.h, [1071](#)
- eWDS
  - common.h, [1071](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID-  
\_IDENT\_FOR\_PROFILE
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT-  
\_DEFINED
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_  
\_PROFILES
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFIL-  
\_E\_FAMILY
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_F-  
LAG\_SET
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_N-  
OT\_SET
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_END
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMI-  
LY
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_HNDL
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_IDENT
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_OP
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_PROFILE\_NUM
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_PROFILE\_TYPE
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_  
\_SUBS\_ID
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LEN\_IN-  
VALID
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NO-  
T\_INITED
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL
  - qmerrno.h, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END
  - qmerrno.h, [1553](#)
- ECIOThresListLen
  - ECIOThresh, [211](#)
- ECIOThresh, [211](#)
  - ECIOThresListLen, [211](#)
  - pECIOThresList, [211](#)
- ECTCallState
  - ECTNum, [212](#)
- ECTNum, [212](#)
  - ECTCallState, [212](#)
  - number, [212](#)
  - presentationInd, [212](#)
- eDevState
  - qaGobiApiCbk.h, [1166](#)
- eDevice
  - sGetDeviceSeriesResult, [707](#)
- eGetDeviceSeries
  - qaGobiApiFms.h, [1290](#)
- eGobiDeviceSeries
  - qaGobiApiFms.h, [1287](#)
- eGobiImageCarrier
  - qaGobiApiFms.h, [1287](#)
- eGobiImageGPS
  - qaGobiApiFms.h, [1289](#)
- eGobiImageRegion
  - qaGobiApiFms.h, [1289](#)
- eGobiImageTech
  - qaGobiApiFms.h, [1289](#)
- eLOG\_LEVEL
  - common.h, [1071](#)
- EMTlv
  - NASQmiCbkNasSystemSelPrefInd, [492](#)
- eQCWWANError
  - qmerrno.h, [1548](#)

- eQMI\_SVC
  - common.h, 1071
- eQMISARRFState
  - qaGobiApiSar.h, 1389
- eQaQMIService
  - qaGobiApiCbk.h, 1200
- ERIFileparams, 213
  - pFile, 213
  - pFileSize, 213
  - qaGobiApiDms.h, 1251
- eSMSEventType
  - qaGobiApiCbk.h, 1167
- ESNString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 848
- eSYS\_SRV\_DOMAIN
  - qaGobiApiNas.h, 1333
- ETWSPLMNInfo
  - eTWSPLMNInfoTlv, 215
- eTWSPLMNInfoTlv, 214
  - ETWSPLMNInfo, 215
  - TlvPresent, 215
- ETWSPLMNTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 939
- ETWSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 939
- eTimeout
  - common.h, 1072
- EVENT\_MASK\_CARD
  - qaGobiApiCbk.h, 1164
- eValid
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- EarMute
  - GetAudioProfileResp, 234
  - GetM2MAudioProfileResp, 251
  - GetM2MAVMuteResp, 253
  - SetAudioProfileReq, 686
  - SetM2MAVMuteReq, 697
- earfcn
  - infoInterFreq, 302
  - LTEInfoIntrafreq, 365
  - ltePCI, 368
  - nas\_infoInterFreq, 416
  - nas\_LTEInfoIntrafreq, 422
  - nas\_umtsLTENbrCell, 460
  - umtsLTENbrCell, 835
- earfcn0
  - lteEARFCN, 359
- earfcn1
  - lteEARFCN, 360
- ecio
  - CDMASSInfo, 149
  - cdmaSSInfo, 149
  - ecioListElement, 211
  - HDRSSInfo, 279
  - hdrSSInfo, 280
  - nas\_ecioListElement, 401
  - nas\_UMTSInfo, 458
  - rxInfo, 668
  - TDSCDMASigInfoExt, 778
  - tdscdmaSigInfoExt, 778
  - UMTSInfo, 833
- ecioDelta
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- ecioInfo
  - nas\_SLQSSignalStrengthsInformation, 449
  - SLQSSignalStrengthsInformation, 728
- ecioList
  - slqsSignalStrengthInfo, 725
  - unpack\_nas\_SLQSGetSignalStrength\_t, 898
- ecioListElement, 210
  - ecio, 211
  - radiolf, 211
- ecioListLen
  - slqsSignalStrengthInfo, 725
  - unpack\_nas\_SLQSGetSignalStrength\_t, 898
- ecioThresholdList
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- ecioThresholdListLen
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- egprsSupp
  - GSMSysInfo, 271
  - nas\_GSMSysInfo, 410
- egprsSuppValid
  - GSMSysInfo, 271
  - nas\_GSMSysInfo, 410
- elevation
  - satelliteInfo, 674
- EmerMode
  - NASEmergencyModeTlv, 469
- emmConnState
  - LTEInfo, 363
  - nas\_LTEInfo, 419
- emmState
  - LTEInfo, 363
  - nas\_LTEInfo, 419
- emmSubState
  - LTEInfo, 363
  - nas\_LTEInfo, 419
- Enable
  - SetM2MAudioLPBKReq, 695
- enable
  - pack\_qos\_SLQSSetQosEventCallback\_t, 551
- enabled
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- EncryptProt
  - protocolSubtypeElement, 613
- EncryptedPIN1
  - pack\_uim\_ChangePin\_t, 560
  - pack\_uim\_SetPinProtection\_t, 562
  - pack\_uim\_UnblockPin\_t, 565
- encryptedPIN1, 212
  - pin1Len, 213

- pin1Val, [213](#)
- EndProactiveSession
  - CatEndProactiveSessionTlv, [138](#)
- EngineState
  - GPSSStateInfo, [265](#)
- engineState
  - QmiCbkLocEngineStateInd, [626](#)
  - unpack\_loc\_EngineState\_Ind\_t, [875](#)
- entries
  - t\_Sv, [776](#)
- eqmiCbkSetStatus
  - sms.h, [1565](#)
- error
  - unpack\_wds\_GetLastMobileIPError\_t, [957](#)
- errorClass
  - SMSAsyncRawSend\_s, [732](#)
- errorRate
  - errorRateListElement, [214](#)
  - nas\_errorRateListElement, [402](#)
- errorRateInfo
  - nas\_SLQSSignalStrengthsInformation, [449](#)
  - SLQSSignalStrengthsInformation, [729](#)
- errorRateList
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- errorRateListElement, [213](#)
  - errorRate, [214](#)
  - radiolf, [214](#)
- errorRateListLen
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- errorState
  - slotInf, [712](#)
  - slotInfo, [714](#)
  - uim\_slotInfo, [807](#)
- esn
  - unpack\_dms\_GetSerialNumbers\_t, [853](#)
- esnSize
  - serialNumbersInfo, [677](#)
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [848](#)
- EspSpi
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- EtwsMessageInfo
  - sMSEtwsMessageTlv, [735](#)
- event
  - unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, [927](#)
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [928](#)
- event\_Index
  - QmiCbkCatEventStatusReportInd, [619](#)
- EventID
  - CatCommonEventTlv, [137](#)
- EventLength
  - CatCommonEventTlv, [137](#)
- eventMask
  - CATEventDataType, [138](#)
  - pack\_uim\_SLQSUIMEventRegister\_t, [563](#)
  - UIMEventRegisterReqResp, [814](#)
  - unpack\_uim\_SLQSUIMEventRegister\_t, [952](#)
- eventRegister
  - LOCEventRegisterReqResp, [350](#)
  - pack\_loc\_EventRegister\_t, [529](#)
- eventType
  - unpack\_swioma\_SLQSOMADMAAlertCallback\_ind\_t, [944](#)
- evrcCapability
  - prefVoiceSO, [600](#)
- executingImage
  - FMSImageIDEntries, [224](#)
  - ImageIDEntries, [286](#)
- exponent
  - pktErrRate, [594](#)
  - unpack\_qos\_pktErrRate\_t, [920](#)
- extBit
  - calledPartySubAdd, [125](#)
- extDispInfo
  - extDispRecInfo, [215](#)
- extDispInfoLen
  - extDispRecInfo, [215](#)
- extDispRecInfo, [215](#)
  - dispType, [215](#)
  - extDispInfo, [215](#)
  - extDispInfoLen, [215](#)
- ExtErrorCode
  - PackCreateProfileOut, [579](#)
- extPowerState
  - LOCExtPowerStateReqResp, [350](#)
  - pack\_loc\_SetExtPowerState\_t, [529](#)
- extendedErrorCode
  - unpack\_wds\_SLQSDeleteProfile\_t, [962](#)
- FIRST\_INSTANCE
  - qaGobiApiCbk.h, [1164](#)
- FLOAT
  - SwiDataTypes.h, [1571](#)
- FMSImageElement, [222](#)
  - buildId, [222](#)
  - buildIdLength, [222](#)
  - imageId, [222](#)
  - imageType, [222](#)
- FMSImageIDEntries, [223](#)
  - executingImage, [224](#)
  - imageIDElement, [224](#)
  - imageIDSize, [224](#)
  - imageType, [224](#)
  - maxImages, [224](#)
- FMSImageIDElement, [222](#)
  - buildID, [223](#)
  - buildIDLength, [223](#)
  - failureCount, [223](#)
  - imageID, [223](#)
  - storageIndex, [223](#)
- FMSImageList, [224](#)
  - imageIDEntries, [225](#)
  - listSize, [225](#)
- FMSPrefImageList, [225](#)

- listEntries, [225](#)
- listSize, [225](#)
- FORBIDDEN\_INDEX
  - qaNasPerformNetworkScan.h, [1546](#)
- FOTAUpdate
  - \_SLQSOMADMSettingsReqParams, [73](#)
  - \_SLQSOMADMSettingsReqParams3, [74](#)
  - pack\_swioa\_SLQSOMADMSetSettings\_t, [559](#)
  - unpack\_swioa\_SLQSOMADMGetSettings\_t, [948](#)
- FOTAdownload
  - \_SLQSOMADMSettingsReqParams, [73](#)
  - \_SLQSOMADMSettingsReqParams3, [74](#)
  - pack\_swioa\_SLQSOMADMSetSettings\_t, [559](#)
  - unpack\_swioa\_SLQSOMADMGetSettings\_t, [948](#)
- FSNumber
  - FactorySequenceNumber, [216](#)
- FactorySequenceNumber, [215](#)
- FSNumber, [216](#)
- failureCount
  - FMSImageIdElement, [223](#)
  - ImageIdElement, [285](#)
- failureReason
  - ssdatasession\_params, [751](#)
- failureReasonv4
  - ssdatasession\_params, [751](#)
- failureReasonv6
  - ssdatasession\_params, [751](#)
- family
  - pack\_wds\_GetDefaultProfileNum\_t, [567](#)
  - pack\_wds\_SetDefaultProfileNum\_t, [569](#)
- feature
  - depersonalizationInformation, [195](#)
  - personalizationStatus, [588](#)
- fileAttributes, [216](#)
  - fileID, [219](#)
  - fileSize, [219](#)
  - fileType, [219](#)
  - rawLen, [219](#)
  - rawValue, [219](#)
  - recordCount, [219](#)
  - recordSize, [219](#)
  - secActivate, [219](#)
  - secActivateMask, [219](#)
  - secDeactivate, [219](#)
  - secDeactivateMask, [219](#)
  - secIncrease, [219](#)
  - secIncreaseMask, [219](#)
  - secRead, [219](#)
  - secReadMask, [219](#)
  - secWrite, [219](#)
  - secWriteMask, [219](#)
- fileID
  - fileAttributes, [219](#)
  - fileInfo, [220](#)
  - uim\_fileInfo, [802](#)
- fileIndex
  - pack\_uim\_ReadTransparent\_t, [561](#)
  - UIMGetFileAttributesReq, [816](#)
  - UIMReadTransparentReq, [820](#)
- fileInfo, [219](#)
  - fileID, [220](#)
  - path, [220](#)
  - pathLen, [220](#)
- fileSize
  - fileAttributes, [219](#)
- fileType
  - fileAttributes, [219](#)
- fill\_pack\_ctx
  - common.h, [1072](#)
- fill\_sdu\_hdr
  - common.h, [1072](#)
- filterId
  - LibPackTFTIDParams, [332](#)
  - TFTIDParams, [782](#)
- Firmware Management Service (FMS), [40](#)
- FirmwareID
  - fwinfo\_s, [226](#)
- FirmwareUpdatStat, [220](#)
  - plmgType, [221](#)
  - pLogString, [221](#)
  - pLogStringLength, [222](#)
  - pRefData, [222](#)
  - pRefString, [222](#)
  - pRefStringLength, [222](#)
  - ResCode, [222](#)
- fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [556](#)
  - SwiLocGetAutoStartResp, [758](#)
  - SwiLocSetAutoStartReq, [759](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- fix\_rate\_reported
  - SwiLocGetAutoStartResp, [758](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [556](#)
  - SwiLocGetAutoStartResp, [758](#)
  - SwiLocSetAutoStartReq, [759](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- fix\_type\_reported
  - SwiLocGetAutoStartResp, [758](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- flags
  - sensorData, [675](#)
- flowLabel
  - LibPackTFTIDParams, [332](#)
  - TFTIDParams, [782](#)
- fms.h, [1107](#)
  - GetValidFwPriCombinations, [1108](#)
  - pack\_fms\_GetImagesPreference, [1109](#)
  - pack\_fms\_GetStoredImages, [1109](#)
  - pack\_fms\_SetImagesPreference, [1109](#)
  - unpack\_fms\_GetImagesPreference, [1109](#)
  - unpack\_fms\_GetStoredImages, [1110](#)
  - unpack\_fms\_SetImagesPreference, [1110](#)
- Forbidden
  - nas\_QmiNas3GppNetworkInfo, [440](#)

- SlqsNas3GppNetworkInfo, 718
- ForceRev0
  - unpack\_nas\_GetCDMANetworkParameters\_t, 885
- ForceXTRADownload
  - qaGobiApiPds.h, 1367
- format
  - SMSTransferRouteMTMessage, 746
  - sMSTransferRouteMTMessage, 745
- ForwardMac
  - protocolSubtypeElement, 613
- fqdnAddr
  - PCSCFFQDNAddress, 581
  - wds\_PCSCFFQDNAddress, 1046
- fqdnLen
  - PCSCFFQDNAddress, 582
  - wds\_PCSCFFQDNAddress, 1046
- freeSlots
  - smsMaxStorageSizeResp, 739
- freq
  - nas\_PhyCaAggPcellInfo, 434
  - nas\_PhyCaAggScellIndType, 436
  - nas\_PhyCaAggScellInfo, 439
  - NASPhyCaAggPcellInfo, 485
  - NASPhyCaAggScellIndType, 487
  - NASPhyCaAggScellInfo, 488
  - PhyCaAggPcellInfo, 589
  - PhyCaAggScellIndType, 591
  - PhyCaAggScellInfo, 593
- freqsLen
  - LTEInfoInterfreq, 363
  - LTEInfoNeighboringGSM, 366
  - LTEInfoNeighboringWCDMA, 367
  - nas\_LTEInfoInterfreq, 420
  - nas\_LTEInfoNeighboringGSM, 423
  - nas\_LTEInfoNeighboringWCDMA, 424
- fromServiceId
  - BroadcastConfig, 121
- fumoResultCode
  - omaDmFotaTlvExt, 518
- function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 556
  - SwiLocGetAutoStartResp, 758
  - SwiLocSetAutoStartReq, 759
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 943
- function\_reported
  - SwiLocGetAutoStartResp, 758
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 943
- FwAutoCheck
  - unpack\_swiodma\_SLQSOMADMGetSettings\_t, 948
- FwAvailability
  - unpack\_swiodma\_SLQSOMADMStartSession\_t, 948
- fwloadsize
  - omaDmFotaTlv, 515
  - unpack\_omaDmFotaTlv\_t, 917
- fwinfo\_s, 225
  - Carrier, 226
  - FirmwareID, 226
  - GPSCapability, 226
  - Region, 226
  - Technology, 226
- fwloadComplete
  - omaDmFotaTlv, 515
  - unpack\_omaDmFotaTlv\_t, 917
- fwvers
  - CurrentImgList, 174
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 864
- g
  - qmifwinfo\_s, 641
- gDIBitRate
  - LibPackQosClassID, 330
  - QosClassID, 649
- GERANInfo, 226
  - arfcn, 227
  - bsic, 227
  - cellID, 227
  - insNmrCellInfo, 227
  - lac, 228
  - nmlnst, 228
  - plmn, 228
  - rxLev, 228
  - timingAdvance, 228
- GPRSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 966
- GPRSQoS, 260
  - delayClass, 261
  - meanThroughputClass, 261
  - peakThroughputClass, 261
  - precedenceClass, 261
  - reliabilityClass, 261
- GPRSRequestedQoS, 261
  - delayClass, 262
  - meanThroughputClass, 262
  - peakThroughputClass, 262
  - precedenceClass, 262
  - reliabilityClass, 262
- GPSCapability
  - fwinfo\_s, 226
- GPSLPM
  - pack\_dms\_SetCustFeature\_t, 522
  - unpack\_dms\_GetCustFeature\_t, 845
- GPSSel
  - pack\_dms\_SetCustFeature\_t, 522
  - unpack\_dms\_GetCustFeature\_t, 845
- GPSStateInfo, 262
  - Altitude, 265
  - EngineState, 265
  - glo\_almanac\_sv\_msk, 265
  - glo\_ephemeris\_sv\_msk, 265
  - glo\_health\_sv\_msk, 265
  - glo\_visible\_sv\_msk, 265
  - gps\_almanac\_sv\_msk, 265
  - gps\_ephemeris\_sv\_msk, 265
  - gps\_health\_sv\_msk, 265
  - gps\_visible\_sv\_msk, 265
  - HorizontalUncertainty, 265

- lono\_valid, [265](#)
- Latitude, [265](#)
- Longitude, [265](#)
- sbas\_almanac\_sv\_msk, [265](#)
- sbas\_ephemeris\_sv\_msk, [265](#)
- sbas\_health\_sv\_msk, [265](#)
- sbas\_visible\_sv\_msk, [266](#)
- Time\_uncert\_ms, [266](#)
- TimeStmp\_gps\_week, [266](#)
- TimeStmp\_tow\_ms, [266](#)
- ValidMask, [266](#)
- VerticalUncertainty, [266](#)
- xtra\_start\_gps\_minutes, [266](#)
- xtra\_start\_gps\_week, [266](#)
- xtra\_valid\_duration\_hours, [266](#)
- GSMRSSIThresh, [268](#)
  - GSMRSSIThreshListLen, [268](#)
  - pGSMRSSIThreshList, [268](#)
- GSMRSSIThreshListLen
  - GSMRSSIThresh, [268](#)
  - nas\_GSMRSSIThresh, [407](#)
- GSMSSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, [907](#)
- GSMSrvStatusInfo, [268](#)
  - isPrefDataPath, [269](#)
  - srvStatus, [269](#)
  - trueSrvStatus, [269](#)
- GSMSysInfo, [269](#)
  - cellId, [271](#)
  - cellIdValid, [271](#)
  - dtmSupp, [271](#)
  - dtmSuppValid, [271](#)
  - egprsSupp, [271](#)
  - egprsSuppValid, [271](#)
  - lac, [271](#)
  - lacValid, [272](#)
  - MCC, [272](#)
  - MNC, [272](#)
  - networkIdValid, [272](#)
  - regRejectInfoValid, [272](#)
  - rejCause, [272](#)
  - rejectSrvDomain, [272](#)
  - sysInfoGSM, [272](#)
- gUIBitRate
  - LibPackQosClassID, [330](#)
  - QosClassID, [649](#)
- GWAOPTiv
  - NASQmiCbkNasSystemSelPrefInd, [492](#)
- GWAcqOrderPref
  - NASGWAcqOrderPrefTlv, [477](#)
- GWAddressV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [966](#)
- gcDumpStrLen
  - CrashInfo, [167](#)
- Generator
  - GetAudioProfileReq, [233](#)
  - GetAudioVolTLBConfigReq, [235](#)
  - GetM2MAudioProfileResp, [251](#)
  - GetM2MAudioVolumeReq, [252](#)
  - SetAudioProfileReq, [686](#)
  - SetAudioVolTLBConfigReq, [687](#)
  - SetM2MAudioVolumeReq, [697](#)
- geoSysIdx
  - AddCDMASysInfo, [96](#)
  - AddSysInfo, [96](#)
  - nas\_AddCDMASysInfo, [385](#)
  - nas\_AddSysInfo, [386](#)
- geranArfcn
  - geranInstInfo, [228](#)
  - nas\_geranInstInfo, [405](#)
- geranBsicBcc
  - geranInstInfo, [228](#)
  - nas\_geranInstInfo, [405](#)
- geranBsicNcc
  - geranInstInfo, [228](#)
  - nas\_geranInstInfo, [405](#)
- geranInst
  - nas\_UMTSInfo, [458](#)
  - UMTSInfo, [833](#)
- GeranInstInfo
  - nas\_UMTSInfo, [458](#)
  - UMTSInfo, [833](#)
- geranInstInfo, [228](#)
  - geranArfcn, [228](#)
  - geranBsicBcc, [228](#)
  - geranBsicNcc, [228](#)
  - geranRssi, [228](#)
- geranRssi
  - geranInstInfo, [228](#)
  - nas\_geranInstInfo, [405](#)
- get\_version
  - common.h, [1072](#)
- GetACCOLC
  - qaGobiApiNas.h, [1334](#)
- GetANAAAAAuthenticationStatus
  - qaGobiApiNas.h, [1334](#)
- GetActivationState
  - qaGobiApiDms.h, [1255](#)
- getAllCallInfo
  - arrCallInfo, [110](#)
- getAllCallInformation, [229](#)
  - ALS, [229](#)
  - Callinfo, [229](#)
  - isEmpty, [229](#)
- getAllCallRmtPtyName
  - arrRemotePartyName, [112](#)
- getAllCallRmtPtyName, [229](#)
  - callID, [230](#)
  - RemotePartyName, [230](#)
- getAllCallRmtPtyNum, [230](#)
  - callID, [230](#)
  - RemotePartyNum, [230](#)
- GetAudioPathConfigReq, [230](#)
  - Item, [231](#)
  - Profile, [231](#)
- GetAudioPathConfigResp, [231](#)

- pCodecSTGain, [232](#)
- pDTMFTXGain, [233](#)
- pECMode, [233](#)
- pMICGainSelect, [233](#)
- pNSEnable, [233](#)
- pRXAGCList, [233](#)
- pRXAVCAGCSwitch, [233](#)
- pRXAVCList, [233](#)
- pRXPCMIIRFtr, [233](#)
- pTXAGCList, [233](#)
- pTXAVCSwitch, [233](#)
- pTXGain, [233](#)
- pTXPCMIIRFtr, [233](#)
- GetAudioProfileReq, [233](#)
  - Generator, [233](#)
- GetAudioProfileResp, [233](#)
  - EarMute, [234](#)
  - MicMute, [234](#)
  - Profile, [234](#)
  - Volume, [234](#)
- GetAudioVolTLBConfigReq, [235](#)
  - Generator, [235](#)
  - Item, [235](#)
  - Profile, [235](#)
  - Volume, [235](#)
- GetAudioVolTLBConfigResp, [235](#)
  - ResCode, [236](#)
- GetAutoconnect
  - qaGobiApiWds.h, [1509](#)
- GetByteTotals
  - qaGobiApiWds.h, [1509](#)
- GetCDMANetworkParameters
  - qaGobiApiNas.h, [1335](#)
- getCallFWExtInfo, [236](#)
  - CallFWExtInfo, [236](#)
  - numInstances, [236](#)
- getCallFWInfo, [236](#)
  - CallFWInfo, [237](#)
  - numInstances, [237](#)
- GetConnectionRate
  - qaGobiApiWds.h, [1510](#)
- GetCustomFeatureV2
  - unpack\_dms\_GetCustFeaturesV2\_t, [845](#)
- getCustomFeatureV2, [237](#)
  - pCustSettingInfo, [237](#)
  - pCustSettingList, [237](#)
  - pGetCustomInput, [237](#)
- getCustomInput, [238](#)
  - cust\_id, [238](#)
  - list\_type, [238](#)
- getDUNCallInfoReq, [238](#)
  - Mask, [239](#)
  - pReportChannelRate, [239](#)
  - pReportConnStatus, [239](#)
  - pReportDataBearerTech, [239](#)
  - pReportDormStatus, [239](#)
  - pTransferStatInd, [239](#)
- getDUNCallInfoResp, [240](#)
- pCallEndReason, [242](#)
- pChannelRate, [242](#)
- pConnectionStatus, [242](#)
- pDataBearerTech, [242](#)
- pDormancyStatus, [242](#)
- pLastCallDataBearerTech, [242](#)
- pLastCallRXOKBytesCnt, [242](#)
- pLastCallTXOKBytesCnt, [242](#)
- pMdmCallDurationActive, [242](#)
- pRXOKBytesCount, [242](#)
- pTXOKBytesCount, [242](#)
- GetDataBearerTechnology
  - qaGobiApiWds.h, [1510](#)
- GetDefaultProfile
  - qaGobiApiWds.h, [1511](#)
- GetDefaultProfileLTE
  - qaGobiApiWds.h, [1513](#)
- GetDefaultProfileNum
  - qaGobiApiWds.h, [1514](#)
- GetDeviceCapabilities
  - qaGobiApiDms.h, [1255](#)
- GetDormancyState
  - qaGobiApiWds.h, [1515](#)
- getDyingGaspCfg, [242](#)
  - pDestSMSContent, [243](#)
  - pDestSMSNum, [243](#)
- getDyingGaspStatistics, [243](#)
  - pSMSAttemptedFlag, [243](#)
  - pTimeStamp, [243](#)
- GetErrRateResp, [244](#)
  - pCDMAFrameErrRate, [244](#)
  - pGSMBER, [244](#)
  - pHDRPackErrRate, [244](#)
  - pWCDMABER, [244](#)
- GetFirmwareRevision
  - qaGobiApiDms.h, [1256](#)
- GetFirmwareRevisions
  - qaGobiApiDms.h, [1257](#)
- GetHRPDStatsResp, [244](#)
  - pDRCPParams, [245](#)
  - pPilotSetData, [245](#)
  - pUATI, [245](#)
- GetHardwareRevision
  - qaGobiApiDms.h, [1257](#)
- GetHomeNetwork
  - qaGobiApiNas.h, [1336](#)
- GetHomeNetwork3GPP2
  - qaGobiApiNas.h, [1337](#)
- GetIMSI
  - qaGobiApiDms.h, [1258](#)
- GetIMSSMSConfigParams, [245](#)
  - pPhoneCtxtURI, [246](#)
  - pPhoneCtxtURLen, [246](#)
  - pSMSFormat, [246](#)
  - pSMSOverIPNwInd, [246](#)
  - pSettingResp, [246](#)
- GetIMSUserConfigParams, [246](#)
  - pIMSDomain, [247](#)

- pIMSDomainLen, [247](#)
  - pSettingResp, [247](#)
- GetIMSVoIPConfigResp, [247](#)
  - pAmrMode, [249](#)
  - pAmrOctetAligned, [249](#)
  - pAmrWBMode, [249](#)
  - pAmrWBOctetAligned, [249](#)
  - pAmrWbEnable, [249](#)
  - pMinSessionExpiryTimer, [249](#)
  - pRTPRTCPInactTimer, [249](#)
  - pRingBackTimer, [249](#)
  - pRingingTimer, [249](#)
  - pScrAmrEnable, [249](#)
  - pScrAmrWbEnable, [249](#)
  - pSessionExpiryTimer, [249](#)
  - pSettingResp, [249](#)
- GetIPAddressLTE
  - qaGobiApiWds.h, [1515](#)
- GetImageStore
  - qaGobiApiFms.h, [1291](#)
- GetImagesPreference
  - qaGobiApiFms.h, [1290](#)
- getIndicationRegResp
  - qaGobiApiSms.h, [1393](#)
- GetInstIDResp, [249](#)
  - pIPFamily, [249](#)
  - pInstanceId, [249](#)
- GetLastMobileLError
  - qaGobiApiWds.h, [1516](#)
- GetM2MAVMuteReq, [252](#)
  - Profile, [252](#)
- GetM2MAVMuteResp, [253](#)
  - CwtMute, [253](#)
  - EarMute, [253](#)
  - MicMute, [253](#)
- GetM2MAudioProfileReq, [250](#)
  - pGenerator, [250](#)
- GetM2MAudioProfileResp, [250](#)
  - CwtMute, [251](#)
  - EarMute, [251](#)
  - Generator, [251](#)
  - MicMute, [251](#)
  - Profile, [251](#)
  - Volume, [251](#)
- GetM2MAudioVolumeReq, [251](#)
  - Generator, [252](#)
  - Profile, [252](#)
- GetM2MAudioVolumeResp, [252](#)
  - Level, [252](#)
- GetM2MSpkrGainReq, [253](#)
  - Profile, [254](#)
- GetM2MSpkrGainResp, [254](#)
  - Value, [254](#)
- GetManufacturer
  - qaGobiApiDms.h, [1258](#)
- GetMobileIP
  - qaGobiApiWds.h, [1516](#)
- GetMobileIPProfile
  - qaGobiApiWds.h, [1517](#)
- GetModelID
  - qaGobiApiDms.h, [1259](#)
- getMsgWaitingInfo, [254](#)
  - msgWaitInfo, [255](#)
  - numInstances, [255](#)
- GetNetworkPreference
  - qaGobiApiNas.h, [1339](#)
- GetNetworkTime
  - qaGobiApiDms.h, [1259](#)
- GetNetworkTimeResp, [255](#)
  - p3GPP2TimeInfo, [255](#)
  - p3GPPTIMEInfo, [255](#)
- GetOfflineReason
  - qaGobiApiDms.h, [1260](#)
- GetPDSDDefaults
  - qaGobiApiPds.h, [1367](#)
- GetPDSSState
  - qaGobiApiPds.h, [1368](#)
- GetPRLVersion
  - qaGobiApiDms.h, [1261](#)
- GetPacketStatistics
  - qaGobiApiWds.h, [1518](#)
- GetPacketStatus
  - qaGobiApiWds.h, [1519](#)
- GetPortAutomaticTracking
  - qaGobiApiPds.h, [1369](#)
- GetPower
  - qaGobiApiDms.h, [1261](#)
- GetProfileSettingIn
  - qaGobiApiWds.h, [1506](#)
- GetProfileSettingOut
  - qaGobiApiWds.h, [1506](#)
- GetRFInfo
  - qaGobiApiNas.h, [1340](#)
- GetRegMgrConfigParams, [255](#)
  - pIMSTestMode, [256](#)
  - pPCSCFPort, [256](#)
  - pPriCSCFPortName, [256](#)
  - pPriCSCFPortNameLen, [256](#)
  - pSettingResp, [256](#)
- GetSIPConfigResp, [256](#)
  - pSIPLocalPort, [257](#)
  - pSettingResp, [257](#)
  - pSigCompEnabled, [257](#)
  - pSubscribeTimer, [257](#)
  - pTimerSIPReg, [257](#)
  - pTimerT1, [257](#)
  - pTimerT2, [257](#)
  - pTimerTf, [257](#)
- GetSMSCAddress
  - qaGobiApiSms.h, [1396](#)
- GetSMSWake
  - qaGobiApiRms.h, [1387](#)
- GetSerialNumbers
  - qaGobiApiDms.h, [1261](#)
- GetServiceAutomaticTracking
  - qaGobiApiPds.h, [1369](#)

- GetServingNetwork
  - qaGobiApiNas.h, [1340](#)
- GetServingNetworkCapabilities
  - qaGobiApiNas.h, [1341](#)
- GetSessionDuration
  - qaGobiApiWds.h, [1519](#)
- GetSessionIDResp, [256](#)
  - pSessionIDv4, [256](#)
  - pSessionIDv6, [256](#)
- GetSessionState
  - qaGobiApiWds.h, [1520](#)
- GetSignalStrengths
  - qaGobiApiNas.h, [1342](#)
- GetStoredImages
  - qaGobiApiFms.h, [1291](#)
- getTransLayerInfoResp
  - qaGobiApiSms.h, [1394](#)
- getTransNWRegInfoResp
  - qaGobiApiSms.h, [1394](#)
- GetValidFwPriCombinations
  - fms.h, [1108](#)
- GetVoiceNumber
  - qaGobiApiDms.h, [1262](#)
- GetXTRAAutomaticDownload
  - qaGobiApiPds.h, [1370](#)
- GetXTRANetwork
  - qaGobiApiPds.h, [1370](#)
- GetXTRAValidity
  - qaGobiApiPds.h, [1371](#)
- glo\_almanac\_sv\_msk
  - GPSSStateInfo, [265](#)
- glo\_ephemeris\_sv\_msk
  - GPSSStateInfo, [265](#)
- glo\_health\_sv\_msk
  - GPSSStateInfo, [265](#)
- glo\_visible\_sv\_msk
  - GPSSStateInfo, [265](#)
- globalCellId
  - LTEInfoIntrafreq, [365](#)
  - nas\_LTEInfoIntrafreq, [422](#)
- glog
  - common.h, [1073](#)
- gloglvl
  - common.h, [1073](#)
- GnssData, [258](#)
  - mask, [259](#)
- gnssSvId
  - satelliteInfo, [674](#)
- gnssSvInfoNotification, [259](#)
  - bAltitudeAssumed, [260](#)
  - pSatelliteInfo, [260](#)
- gnssSvUsedList
  - loc\_svUsedforFix, [346](#)
  - svUsedforFix\_s, [755](#)
- gnssSvUsedList\_len
  - loc\_svUsedforFix, [346](#)
  - svUsedforFix\_s, [755](#)
- Gpp2TimeZone
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- GppNetworkDSTAdjustment
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- GppTimeZone
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- gps\_almanac\_sv\_msk
  - GPSSStateInfo, [265](#)
- gps\_ephemeris\_sv\_msk
  - GPSSStateInfo, [265](#)
- gps\_health\_sv\_msk
  - GPSSStateInfo, [265](#)
- gps\_visible\_sv\_msk
  - GPSSStateInfo, [265](#)
- GpsEnable
  - custFeaturesInfo, [181](#)
  - pack\_dms\_SetCustFeature\_t, [522](#)
  - unpack\_dms\_GetCustFeature\_t, [845](#)
- gpsTime
  - qaGobiApiCbk.h, [1167](#)
- gpsTime\_s, [266](#)
  - gpsTimeOfWeekMs, [266](#)
  - gpsWeek, [266](#)
- gpsTimeOfWeekMs
  - gpsTime\_s, [266](#)
  - loc\_gpsTime, [341](#)
  - t\_gpsTime, [775](#)
- gpsWeek
  - gpsTime\_s, [266](#)
  - loc\_gpsTime, [341](#)
  - t\_gpsTime, [775](#)
- grntDownlinkBitrate
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [837](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- grntUplinkBitrate
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [838](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- gsmAmrStat
  - curAMRConfig, [171](#)
- GsmCellInfo
  - IteGsmCellInfo, [361](#)
  - nas\_IteGsmCellInfo, [417](#)
- gsmCellInfo, [266](#)
  - arfcn, [267](#)
  - band1900, [267](#)
  - bsicId, [267](#)
  - cellIdValid, [267](#)
  - rssI, [267](#)
  - srxlev, [267](#)
- gsmUmtsDI
  - NasSwilndReg, [497](#)

- pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 543
- gsmUmtsUI
  - NasSwiIndReg, 497
  - pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 544
- guaranteedRate
  - dataRate, 190
  - unpack\_qos\_dataRate\_t, 918
- gwAddressV6
  - IPv6GWAddressInfo, 306
  - wds\_IPV6GWAddressInfo, 1046
- gwV6PrefixLen
  - IPv6GWAddressInfo, 306
  - wds\_IPV6GWAddressInfo, 1046
- gyroAcceptReady
  - qaGobiApiCbk.h, 1167
- gyroAcceptReady\_s, 272
  - batchPerSec, 272
  - injectEnable, 272
  - samplesPerBatch, 273
- gyroTempAcceptReady
  - qaGobiApiCbk.h, 1167
- gyroTempAcceptReady\_s, 273
  - batchPerSec, 273
  - injectEnable, 273
  - samplesPerBatch, 273
- HASPI
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- HASState
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- HDOP
  - loc\_precisionDilution, 343
  - precisionDilution\_s, 598
- HDRECIOTresh, 273
  - HDRECIOTreshListLen, 274
  - pHDRECIOTreshList, 274
- HDRECIOTreshListLen
  - HDRECIOTresh, 274
  - nas\_HDRECIOTresh, 411
- HDRIOTresh, 274
  - HDRIOTreshListLen, 274
  - pHDRIOTreshList, 274
- HDRIOTreshListLen
  - HDRIOTresh, 274
  - nas\_HDRIOTresh, 412
- HDRPersonalityInd, 274
  - pCurrentPersonality, 275
  - pPersonalityListLength, 275
  - pProtocolSubtypeElement, 275
- HDRPersonalityResp, 275
  - pCurrentPersonality, 275
  - pPersonalityListLength, 275
  - pProtocolSubtypeElement, 275
- HDRProtSubtypResp, 275
  - pAppSubType, 276
  - pCurrentPrsnlty, 276
  - pPersonalityListLength, 276
- pProtoSubTypElmnt, 276
- HDRRSSITresh, 276
  - HDRRSSITreshListLen, 277
  - pHDRRSSITreshList, 277
- HDRRSSITreshListLen
  - HDRRSSITresh, 277
  - nas\_HDRRSSITresh, 412
- HDRSINRThresListLen
  - HDRSINRThresh, 277
- HDRSINRThresh, 277
  - HDRSINRThresListLen, 277
  - pHDRSINRThresList, 277
- HDRSINRThreshListLen
  - HDRSINRThreshold, 278
  - nas\_HDRSINRThreshold, 413
- HDRSINRThreshold, 277
  - HDRSINRThreshListLen, 278
  - pHDRSINRThresList, 278
- HDRSSInfo, 278
  - ecio, 279
  - io, 279
  - rssI, 279
  - sinr, 279
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 907
- HDRSysInfo, 280
  - hdrActiveProt, 282
  - hdrActiveProtValid, 282
  - hdrPersonality, 282
  - hdrPersonalityValid, 282
  - is856SysId, 282
  - is856SysIdValid, 282
  - isSysPrIMatch, 282
  - isSysPrIMatchValid, 282
  - sysInfoHDR, 282
- HWVersion
  - DeviceConfigDetail, 197
- HardwareControlledMode
  - unpack\_dms\_GetPower\_t, 852
- hdrActiveProt
  - HDRSysInfo, 282
  - nas\_HDRSysInfo, 415
- hdrActiveProtValid
  - HDRSysInfo, 282
  - nas\_HDRSysInfo, 415
- hdrHybrid
  - detailSvcInfo, 196
  - nas\_detailSvcInfo, 401
- HdrPersonality
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- hdrPersonality
  - HDRSysInfo, 282
  - nas\_HDRSysInfo, 415
  - NASServingSystemInfo, 494
  - qaQmiServingSystemParam, 619
  - ServingSystemInfo, 680
- hdrPersonalityValid
  - HDRSysInfo, 282
  - nas\_HDRSysInfo, 415

- hdrSSInfo, 279
  - ecio, 280
  - io, 280
  - rsi, 280
  - sinr, 280
- hdrSrvStatus
  - detailSvcInfo, 196
  - nas\_detailSvcInfo, 401
- healthStatus
  - satelliteInfo, 674
- helper\_get\_resp\_ctx
  - common.h, 1072
- helper\_get\_xid
  - common.h, 1073
- helper\_set\_log\_func
  - common.h, 1073
- helper\_set\_log\_lvl
  - common.h, 1073
- homeOrigVoiceSO
  - prefVoiceSO, 600
- homePageVoiceSO
  - prefVoiceSO, 600
- homeSIDNID, 282
  - numInstances, 282
  - SidNid, 282
- HorizontalUncertainty
  - GPSSStateInfo, 265
- hotSwap
  - hotSwapStatus, 283
  - uim\_hotSwapStatus, 802
- hotSwapLength
  - hotSwapStatus, 283
  - uim\_hotSwapStatus, 802
- hotSwapStatus, 282
  - hotSwap, 283
  - hotSwapLength, 283
- hour
  - nas\_timeInfo, 456
  - nas\_UniversalTime, 461
  - timeInfo, 784
  - UniversalTime, 843
- hsCallStatus
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- hsCallStatusValid
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- hsInd
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- hsIndValid
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- hwType
  - WdsDHCPv4HWConfig, 1057
  - wdsDhcpv4HwConfig, 1056
- hwVer
  - unpack\_dms\_GetHardwareRevision\_t, 851
- iFaceTab
  - PCMparams, 581
- iFaceTabLen
  - PCMparams, 581
- iGetByteTotals
  - qaGobiApiWds.h, 1520
- iGetConnectionRate
  - qaGobiApiWds.h, 1521
- iGetPacketStatistics
  - qaGobiApiWds.h, 1521
- iLTEbandValue
  - nas\_PhyCaAggPcellInfo, 434
  - nas\_PhyCaAggScellInfo, 439
  - NASPhyCaAggPcellInfo, 485
  - NASPhyCaAggScellInfo, 488
  - PhyCaAggPcellInfo, 589
  - PhyCaAggScellInfo, 593
- IMCNflag
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 966
- IMEIString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 848
- IMG\_ID\_LEN
  - qaGobiApiFms.h, 1287
- IMGDETAILS\_LEN
  - qaGobiApiDms.h, 1247
- IMS Service (IMS), 49
- IMSALndRegisterInfo, 286
  - pPdpStatusConfig, 287
  - pRatHandoverStatusConfig, 287
  - pRegStatusConfig, 287
  - pServiceStatusConfig, 287
- IMSARegistrationStatus, 289
  - plmsRegErrCode, 290
  - plmsRegStatus, 290
  - pNewlmsRegStatus, 290
- IMSAServiceStatus, 290
  - pSmsServiceRat, 292
  - pSmsServiceStatus, 292
  - pUtServiceRat, 292
  - pUtServiceStatus, 292
  - pVoipServiceRat, 292
  - pVoipServiceStatus, 292
  - pVsServiceRat, 292
  - pVsServiceStatus, 292
  - pVtServiceRat, 292
  - pVtServiceStatus, 292
- IMSASupportedFieldsResp, 292
  - pIndFieldsList, 293
  - pReqFieldsList, 293
  - pRespFieldsList, 293
- IMSASupportedMsgInfo, 293
  - pSupportedMsgList, 293
- IMSI\_M\_S1\_LENGTH
  - qaGobiApiNas.h, 1321
- IMSI\_M\_S2\_LENGTH
  - qaGobiApiNas.h, 1321
- IMSInfo
  - sMSOnIMSTlv, 743

- IMSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [939](#)
- INDEX\_ZERO
  - qaNasPerformNetworkScan.h, [1546](#)
- INT32
  - SwiDataTypes.h, [1571](#)
- INT8
  - SwiDataTypes.h, [1571](#)
- INVALID\_INSTACNE
  - qaGobiApiCbk.h, [1164](#)
- IOThresListLen
  - IOThresh, [303](#)
- IOThresh, [303](#)
  - IOThresListLen, [303](#)
  - pIOThresList, [303](#)
- IPAddress
  - DataStatusDetail, [192](#)
- IPAddressV6
  - IPV6AddressInfo, [305](#)
  - ipv6AddressInfo, [305](#)
  - wds\_IPV6AddressInfo, [1045](#)
- IPFamSupport
  - pack\_dms\_SetCustFeature\_t, [522](#)
  - unpack\_dms\_GetCustFeature\_t, [845](#)
- IPFamilyPreference
  - pack\_wds\_SLQSSetIPFamilyPreference\_t, [576](#)
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [966](#)
- IPSECSPi
  - LibPackTFTIDParams, [332](#)
  - TFTIDParams, [782](#)
- IPv4
  - qaGobiApiCbk.h, [1164](#)
- IPv4V6
  - qaGobiApiCbk.h, [1164](#)
- IPv6
  - qaGobiApiCbk.h, [1164](#)
- IPv6AddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- IPv6AddressInfo, [304](#)
  - IPAddressV6, [305](#)
  - IPv6PrefixLen, [305](#)
- IPv6GWAddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- IPv6GWAddressInfo, [305](#)
  - gwAddressV6, [306](#)
  - gwV6PrefixLen, [306](#)
- IPv6PrefixLen
  - IPv6AddressInfo, [305](#)
  - ipv6AddressInfo, [305](#)
  - wds\_IPV6AddressInfo, [1045](#)
- IPv4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [966](#)
- IPv4Addr, [303](#)
  - addr, [304](#)
  - subnetMask, [304](#)
- IPv4DstAddr
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv4SrcAddr
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv4Tos
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv6Addr, [304](#)
  - addr, [304](#)
  - prefixLen, [304](#)
- IPv6DstAddr
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv6Label
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv6SrcAddr
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- IPv6TrafCls, [306](#)
  - mask, [306](#)
  - unpack\_qos\_swiQosFilter\_t, [931](#)
  - val, [306](#)
- iSLQSMISetIPFamilyPreference
  - qaGobiApiWds.h, [1521](#)
- iSLQSSetDUNCallInfoCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetSignalStrengthsCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetWdsFirstInstEventCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetWdsSecondInstEventCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetWdsThirdInstEventCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetWdsXferStatsFirstInstCallback
  - qaGobiApiCbk.h, [1201](#)
- iSLQSSetWdsXferStatsSecondInstCallback
  - qaGobiApiCbk.h, [1201](#)
- iSetCATEventCallback
  - qaGobiApiCbk.h, [1201](#)
- iSetSignalStrengthCallback
  - qaGobiApiCbk.h, [1201](#)
- Id
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- id
  - BdsSV, [119](#)
  - CSGID, [170](#)
  - loc\_BdsSV, [336](#)
  - loc\_SV, [344](#)
  - nas\_CSGID, [397](#)
  - QosFlowInfoState, [652](#)
  - SV, [754](#)
  - swiQosModifyReq, [771](#)
  - unpack\_qos\_QosFlowInfoState\_t, [923](#)
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [928](#)
- id\_length
  - custSettingInfo, [184](#)
  - DMScustSettingInfo, [202](#)
- IdleState
  - protocolSubtypeElement, [613](#)
- image\_info\_t, [283](#)
  - buildID, [283](#)
  - buildIDLen, [283](#)

- imageType, 283
- uniqueID, 283
- ImageElement, 283
  - buildId, 284
  - buildIdLength, 284
  - imageId, 284
  - imageType, 284
- imageID
  - FMSImageIdElement, 223
  - ImageIdElement, 285
- imageIDElement
  - FMSImageIDEntries, 224
  - ImageIDEntries, 286
- ImageIDEntries, 285
  - executingImage, 286
  - imageIDElement, 286
  - imageIDSize, 286
  - imageType, 286
  - maxImages, 286
- imageIDEntries
  - FMSImageList, 225
  - ImageList, 286
- imageIDSize
  - FMSImageIDEntries, 224
  - ImageIDEntries, 286
- imageId
  - FMSImageElement, 222
  - ImageElement, 284
- ImageIdElement, 284
  - buildID, 285
  - buildIDLength, 285
  - failureCount, 285
  - imageID, 285
  - storageIndex, 285
- ImageList, 286
  - imageIDEntries, 286
  - listSize, 286
- imageList
  - unpack\_fms\_GetStoredImages\_t, 868
- ImageListSize
  - unpack\_fms\_GetImagesPreference\_t, 867
- imageListSize
  - pack\_fms\_SetImagesPreference\_t, 526
- imageType
  - CurrImageInfo, 176
  - FMSImageElement, 222
  - FMSImageIDEntries, 224
  - image\_info\_t, 283
  - ImageElement, 284
  - ImageIDEntries, 286
- ImageTypes
  - unpack\_fms\_SetImagesPreference\_t, 868
- ImageTypesSize
  - unpack\_fms\_SetImagesPreference\_t, 868
- imagelistSize
  - unpack\_fms\_GetStoredImages\_t, 868
- imei\_no
  - unpack\_dms\_GetSerialNumbers\_t, 853
- imeiSize
  - serialNumbersInfo, 677
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 848
- imeiSvnSize
  - serialNumbersInfo, 678
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 848
- ImeiSvnString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 848
- imeisv\_svn
  - unpack\_dms\_GetSerialNumbers\_t, 853
- imgType
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 865
- imsCfgIndRegisterInfo, 294
  - pRegMgrConfigEvents, 295
  - pSIPConfigEvents, 295
  - pSMSConfigEvents, 295
  - pUserConfigEvents, 295
  - pVoIPConfigEvents, 296
- imsRegMgrConfigInfo, 296
  - pCSCFPortName, 296
  - pIMSTestMode, 296
  - pPriCSCFPort, 296
- imsRegState
  - CommInfo, 163
  - nas\_CommInfo, 397
- imsSIPConfigInfo, 296
  - pSIPLocalPort, 297
  - pSigCompEnabled, 297
  - pSubscribeTimer, 297
  - pTimerSIPReg, 297
  - pTimerT1, 297
  - pTimerT2, 297
  - pTimerTf, 297
- imsSMSConfigInfo, 297
  - pPhoneCtxtURI, 298
  - pSMSFormat, 298
  - pSMSOverIPNwInd, 298
- imsUserConfigInfo, 298
  - pIMSDomain, 298
- imsVoIPConfigInfo, 299
  - pAmrMode, 301
  - pAmrOctetAligned, 301
  - pAmrWBMode, 301
  - pAmrWBOctetAligned, 301
  - pAmrWbEnable, 301
  - pMinSessionExpiryTimer, 301
  - pRTPRTCPInactTimer, 301
  - pRingBackTimer, 301
  - pRingingTimer, 301
  - pScrAmrEnable, 301
  - pScrAmrWbEnable, 301
  - pSessionExpiryTimer, 301
- imsaPdpStatusInfo, 287
  - connetionState, 288
  - pFailErrorCode, 288
- imsaRatStatusInfo, 288
  - pErrorCodeStr, 289
  - pRATStatus, 289

- pSrcRAT, [289](#)
  - pTgtRAT, [289](#)
- imsaRegStatusInfo, [290](#)
  - plmsRegStatus, [290](#)
  - pRegStatusErrorCode, [290](#)
  - pbIMSRegistered, [290](#)
- imsaSvcStatusInfo, [293](#)
  - pSMSSvcRAT, [294](#)
  - pSMSSvcStatus, [294](#)
  - pUTSvcRAT, [294](#)
  - pUTSvcStatus, [294](#)
  - pVOIPSvcRAT, [294](#)
  - pVOIPSvcStatus, [294](#)
  - pVTSvcRAT, [294](#)
  - pVTSvcStatus, [294](#)
- imsi
  - unpack\_dms\_GetIMSI\_t, [851](#)
- imsi\_11\_12
  - CDMASysInfoExt, [153](#)
  - nas\_CDMASysInfoExt, [394](#)
- imsiM1112
  - minBasedIMSI, [380](#)
- imsiMS1
  - minBasedIMSI, [380](#)
- imsiMS2
  - minBasedIMSI, [380](#)
- imsiT1112
  - trueIMSI, [792](#)
- imsiTS1
  - trueIMSI, [792](#)
- imsiTS2
  - trueIMSI, [792](#)
- imsiTaddrNum
  - trueIMSI, [792](#)
- InUse
  - nas\_QmiNas3GppNetworkInfo, [440](#)
  - SlqsNas3GppNetworkInfo, [718](#)
- includes\_pcs\_digit
  - nas\_QmisNasPcsDigit, [441](#)
  - SlqsNasPcsDigit, [719](#)
- IndFieldsList, [301](#)
  - indicationFields, [301](#)
  - indicationFieldsLen, [301](#)
- index
  - pack\_wds\_GetMobileIPProfile\_t, [567](#)
  - pack\_wds\_SetDefaultProfileNum\_t, [569](#)
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
  - swiQosFilter, [766](#)
  - swiQosFlow, [769](#)
  - swiQosReq, [772](#)
  - unpack\_qos\_swiQosFilter\_t, [931](#)
  - unpack\_qos\_swiQosFlow\_t, [935](#)
  - unpack\_wds\_GetDefaultProfileNum\_t, [957](#)
- index1xPri
  - cardStatus, [135](#)
  - uim\_cardStatus, [800](#)
- index1xSec
  - cardStatus, [135](#)
- uim\_cardStatus, [800](#)
- indexGwPri
  - cardStatus, [135](#)
  - uim\_cardStatus, [800](#)
- indexGwSec
  - cardStatus, [135](#)
  - uim\_cardStatus, [800](#)
- indicationFields
  - IndFieldsList, [301](#)
- indicationFieldsLen
  - IndFieldsList, [301](#)
- Info
  - unpack\_nas\_SLQSNasSwiOTAMessageCallback-\_ind\_t, [910](#)
  - unpack\_nas\_SLQSSetSysSelectionPrefCallBack-\_ind\_t, [910](#)
- infoInterFreq, [302](#)
  - cell\_resele\_priority, [302](#)
  - cellInterFreqParams, [302](#)
  - cells\_len, [302](#)
  - earfcn, [302](#)
  - threshXHigh, [303](#)
  - threshXLow, [303](#)
- InfoInterfreq
  - LTEInfoInterfreq, [363](#)
  - nas\_LTEInfoInterfreq, [420](#)
- InitiateDomainAttach
  - qaGobiApiNas.h, [1343](#)
- InitiateNetworkRegistration
  - qaGobiApiNas.h, [1343](#)
- injectEnable
  - accelAcceptReady\_s, [93](#)
  - accelTempAcceptReady\_s, [94](#)
  - gyroAcceptReady\_s, [272](#)
  - gyroTempAcceptReady\_s, [273](#)
- injectSensorDataStatus
  - QmiCbkLocInjectSensorDataInd, [629](#)
- injectTimeSyncStatus
  - QmiCbkLocInjectTimeInd, [629](#)
- insNmrCellInfo
  - GERANInfo, [227](#)
  - nas\_GERANInfo, [404](#)
- instanceId
  - ssdatasession\_params, [751](#)
- instancesSize
  - unpack\_nas\_GetRFInfo\_t, [887](#)
- interval
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [577](#)
  - TransferStatsDataType, [790](#)
- Io
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSSetSignalStrength\_t, [898](#)
- io
  - HDRSSInfo, [279](#)
  - hdrSSInfo, [280](#)
  - nas\_SLQSSignalStrengthsInformation, [449](#)
  - SLQSSignalStrengthsInformation, [729](#)
- ioDelta

- nas\_SLQSSignalStrengthsIndReq, [448](#)
- SLQSSignalStrengthsIndReq, [727](#)
- lono\_valid
  - GPSSStateInfo, [265](#)
- ip
  - WdsIpAddressInfoReq, [1061](#)
- ipAddress
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
- ipFamily
  - \_packetSrvStatus, [62](#)
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [968](#)
- ipVersion
  - LibPackTFTIDParams, [332](#)
  - TFTIDParams, [782](#)
- ipaddr
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- ipaddrv6
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- ipfamily
  - ssdatasession\_params, [751](#)
- ipv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [972](#)
- ipv4GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [972](#)
- ipv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [972](#)
- ipv6AddressInfo, [305](#)
  - IPAddressV6, [305](#)
  - IPv6PrefixLen, [305](#)
- ipv6GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [972](#)
- is856SysId
  - HDRSysInfo, [282](#)
  - nas\_HDRSysInfo, [415](#)
- is856SysIdValid
  - HDRSysInfo, [282](#)
  - nas\_HDRSysInfo, [415](#)
- is\_DataRate\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_EspSpi\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv4DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv4SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv4Tos\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv6DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv6Label\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv6SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_IPv6TrafCls\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_Id\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_Jitter\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_Latency\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_LteBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, [862](#)
- is\_LteQci\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_MaxAllowedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_MinPolicedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_NxtHdrProto\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_PktErrRate\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_Precedence\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_ProfileId3GPP2\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_RxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- is\_TCPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_TCPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [931](#)
- is\_TdsBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, [862](#)
- is\_TokenBucket\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_TrafficClass\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_TranDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- is\_TranSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- is\_TxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- is\_UDPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- is\_UDPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- is\_val\_3GPP2Pri\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_val\_3GPPImCn\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_val\_3GPPResResidualBER\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_val\_3GPPSigInd\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- is\_val\_3GPPTraHdlPri\_Available
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- isEmpty
  - getAllCallInformation, [229](#)

- isInTraffic
  - txInfo, [793](#)
- isModByCC
  - SUPInfo, [753](#)
- isNewFlow
  - QosFlowInfoState, [652](#)
  - unpack\_qos\_QosFlowInfoState\_t, [923](#)
- isPrefDataPath
  - GSMSrvStatusInfo, [269](#)
  - nas\_GSMSrvStatusInfo, [408](#)
  - nas\_SrvStatusInfo, [450](#)
  - SrvStatusInfo, [749](#)
- isRadioTuned
  - rxInfo, [668](#)
- isSysForbidden
  - detailSvcInfo, [196](#)
  - nas\_detailSvcInfo, [401](#)
  - nas\_sysInfoCommon, [452](#)
  - sysInfoCommon, [775](#)
- isSysForbiddenValid
  - nas\_sysInfoCommon, [452](#)
  - sysInfoCommon, [775](#)
- isSysPriMatch
  - CDMASysInfo, [152](#)
  - HDRSysInfo, [282](#)
  - nas\_CDMASysInfo, [393](#)
  - nas\_HDRSysInfo, [415](#)
- isSysPriMatchValid
  - CDMASysInfo, [152](#)
  - HDRSysInfo, [282](#)
  - nas\_CDMASysInfo, [393](#)
  - nas\_HDRSysInfo, [415](#)
- IsVoiceEnabled
  - pack\_dms\_SetCustFeature\_t, [522](#)
  - unpack\_dms\_GetCustFeature\_t, [845](#)
- Item
  - GetAudioPathConfigReq, [231](#)
  - GetAudioVoITLBConfigReq, [235](#)
  - SetAudioVoITLBConfigReq, [687](#)
- Jitter
  - unpack\_qos\_swiQosFlow\_t, [935](#)
- KeyExchange
  - protocolSubtypeElement, [613](#)
- LIBPACK\_QMI\_CBK\_PARAM\_NOCHANGE
  - sms.h, [1565](#)
- LIBPACK\_QMI\_CBK\_PARAM\_RESET
  - sms.h, [1565](#)
- LIBPACK\_QMI\_CBK\_PARAM\_SET
  - sms.h, [1565](#)
- LBPTiv
  - NASQmiCbkNasSystemSelPrefInd, [492](#)
- LEN
  - qaGobiApiDcs.h, [1238](#)
- LOCEVENTMASKGNSSSVINFO
  - loc.h, [1113](#)
- LOCEVENTMASKNMEA
  - loc.h, [1114](#)
- LOCEVENTMASKWIFIREQ
  - loc.h, [1114](#)
- LOCEventRegisterReqResp, [348](#)
  - eventRegister, [350](#)
- LOCExtPowerStateReqResp, [350](#)
  - extPowerState, [350](#)
- LOCStartReq, [356](#)
  - pApplicationInfo, [357](#)
  - pConfigAltitudeAssumed, [357](#)
  - pHorizontalAccuracyLvl, [357](#)
  - pIntermediateReportState, [358](#)
  - pMinIntervalTime, [358](#)
  - pRecurrenceType, [358](#)
  - SessionId, [358](#)
- LOCStopReq, [358](#)
  - sessionId, [358](#)
- LPCSTR
  - SwiDataTypes.h, [1571](#)
- LTEAttachProfileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [576](#)
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [962](#)
- LTEAttachProfileListLen
  - \_slqs3GPPConfigItem, [66](#)
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [576](#)
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [963](#)
- LTETestBandPref
  - NASLTETestBandPreferenceTlv, [481](#)
- LTETestCphyCAInfo
  - unpack\_nas\_SlqsGetLTETestCphyCAInfo\_t, [894](#)
- LTEInfo, [361](#)
  - band, [363](#)
  - bandwidth, [363](#)
  - emmConnState, [363](#)
  - emmState, [363](#)
  - emmSubState, [363](#)
  - RXChan, [363](#)
  - TXChan, [363](#)
- LTEInfoInterfreq, [363](#)
  - freqsLen, [363](#)
  - InfoInterfreq, [363](#)
  - ueInIdle, [363](#)
- LTEInfoIntrafreq, [363](#)
  - CellParams, [365](#)
  - cellReselPriority, [365](#)
  - cellsLen, [365](#)
  - earfcn, [365](#)
  - globalCellId, [365](#)
  - plmn, [365](#)
  - sIntraSearch, [365](#)
  - sNonIntraSearch, [365](#)
  - servingCellId, [365](#)
  - tac, [365](#)
  - threshServingLow, [365](#)
  - ueInIdle, [365](#)
- LTEInfoNeighboringGSM, [366](#)
  - freqsLen, [366](#)
  - LteGsmCellInfo, [366](#)



- [readTransparentInfo, 654](#)
  - [SMSCAddress, 733](#)
  - [sMSCAddress, 733](#)
  - [SMSEtwsMessage, 735](#)
  - [sMSEtwsMessage, 734](#)
  - [SMSTransferRouteMTMessage, 746](#)
  - [sMSTransferRouteMTMessage, 745](#)
  - [uim\\_readTransparentInfo, 803](#)
- [Level](#)
  - [GetM2MAudioVolumeResp, 252](#)
  - [SetM2MAudioVolumeReq, 697](#)
- [LibPackGPRSRequestedQoS, 306](#)
  - [delayClass, 307](#)
  - [meanThroughputClass, 307](#)
  - [peakThroughputClass, 307](#)
  - [precedenceClass, 307](#)
  - [reliabilityClass, 307](#)
- [LibPackQoSClassID, 329](#)
  - [gDIBitRate, 330](#)
  - [gUIBitRate, 330](#)
  - [maxDIBitRate, 330](#)
  - [maxUIBitRate, 330](#)
  - [QCI, 330](#)
- [LibPackTFTIDParams, 330](#)
  - [destPortRangeEnd, 331](#)
  - [destPortRangeStart, 332](#)
  - [eValid, 332](#)
  - [filterId, 332](#)
  - [flowLabel, 332](#)
  - [IPSECSPi, 332](#)
  - [ipVersion, 332](#)
  - [nextHeader, 332](#)
  - [pSourceIP, 332](#)
  - [sourceIPMask, 332](#)
  - [srcPortRangeEnd, 332](#)
  - [srcPortRangeStart, 332](#)
  - [tosMask, 332](#)
- [LibPackUMTSQoS, 332](#)
  - [deliveryErrSDU, 334](#)
  - [grntDownlinkBitrate, 334](#)
  - [grntUplinkBitrate, 334](#)
  - [maxDownlinkBitrate, 334](#)
  - [maxSDUSize, 334](#)
  - [maxUplinkBitrate, 334](#)
  - [qosDeliveryOrder, 334](#)
  - [resBerRatio, 334](#)
  - [sduErrorRatio, 334](#)
  - [trafficClass, 334](#)
  - [trafficPriority, 334](#)
  - [transferDelay, 334](#)
- [LibPackUMTSReqQoSsigInd, 334](#)
  - [SigInd, 335](#)
  - [UMTSReqQoS, 335](#)
- [LibPackprofile\\_3GPP, 318](#)
  - [pAPNClass, 323](#)
  - [pAPNDisabledFlag, 323](#)
  - [pAPNName, 323](#)
  - [pAPNnameSize, 323](#)
  - [pAddrAllocPref, 323](#)
  - [pAuthenticationPref, 323](#)
  - [pGPRSMinimumQoS, 323](#)
  - [pGPRSRequestedQoS, 323](#)
  - [pIPv4AddrPref, 323](#)
  - [pIPv6AddrPref, 323](#)
  - [pImCnFlag, 323](#)
  - [pPDNInactivTimeout, 323](#)
  - [pPDPTtype, 323](#)
  - [pPassword, 323](#)
  - [pPasswordSize, 323](#)
  - [pPcscfAddrUsingDhcp, 323](#)
  - [pPcscfAddrUsingPCO, 323](#)
  - [pPdpAccessConFlag, 323](#)
  - [pPdpContext, 323](#)
  - [pPdpDataCompType, 323](#)
  - [pPdpHdrCompType, 323](#)
  - [pPriDNSIPv4AddPref, 323](#)
  - [pPriDNSIPv6addpref, 323](#)
  - [pPrimaryID, 323](#)
  - [pProfileName, 324](#)
  - [pProfileNameSize, 324](#)
  - [pQoSClassID, 324](#)
  - [pSecDNSIPv4AddPref, 324](#)
  - [pSecDNSIPv6addpref, 324](#)
  - [pSecondaryFlag, 324](#)
  - [pTFTID1Params, 324](#)
  - [pTFTID2Params, 324](#)
  - [pUMTSMInQoS, 324](#)
  - [pUMTSMInQoSsigInd, 324](#)
  - [pUMTSReqQoS, 324](#)
  - [pUMTSReqQoSsigInd, 324](#)
  - [pUsername, 324](#)
  - [pUsernameSize, 324](#)
- [LibPackprofile\\_3GPP2, 324](#)
  - [pAPNClass3GPP2, 328](#)
  - [pAPNEnabled3GPP2, 328](#)
  - [pAllowLinger, 328](#)
  - [pApnString, 328](#)
  - [pApnStringSize, 328](#)
  - [pAppPriority, 328](#)
  - [pAppType, 328](#)
  - [pAuthPassword, 328](#)
  - [pAuthPassword\\_tSize, 328](#)
  - [pAuthProtocol, 328](#)
  - [pAuthRetryCount, 329](#)
  - [pAuthTimeout, 329](#)
  - [pDataMode, 329](#)
  - [pDataRate, 329](#)
  - [pIpcpAckTimeout, 329](#)
  - [pIpcpCreqRetryCount, 329](#)
  - [pIscscfAddressNedded, 329](#)
  - [pLcpAckTimeout, 329](#)
  - [pLcpCreqRetryCount, 329](#)
  - [pNegoDnsSrvrPref, 329](#)
  - [pPDNInactivTimeout3GPP2, 329](#)
  - [pPdnType, 329](#)
  - [pPppSessCloseTimer1x, 329](#)

- pPppSessCloseTimerDO, [329](#)
- pPriV6DnsAddress, [329](#)
- pPrimaryV4DnsAddress, [329](#)
- pRATType, [329](#)
- pSecV6DnsAddress, [329](#)
- pSecondaryV4DnsAddress, [329](#)
- pUserId, [329](#)
- pUserIdSize, [329](#)
- libpack\_GetVersion
  - common.h, [1073](#)
- libpack\_log
  - common.h, [1073](#)
- LibpackProfile3GPP, [307](#)
  - pAPNClass, [312](#)
  - pAPNDisabledFlag, [312](#)
  - pAPNName, [312](#)
  - pAPNnameSize, [312](#)
  - pAddrAllocPref, [312](#)
  - pAuthenticationPref, [312](#)
  - pGPRSMinimumQoS, [312](#)
  - pGPRSRequestedQos, [312](#)
  - pIPv4AddrPref, [312](#)
  - pIPv6AddPref, [312](#)
  - plmCnFlag, [312](#)
  - pPDNInactivTimeout, [312](#)
  - pPDPtype, [312](#)
  - pPassword, [312](#)
  - pPasswordSize, [312](#)
  - pPcscfAddrUsingDhcp, [312](#)
  - pPcscfAddrUsingPCO, [312](#)
  - pPdpAccessConFlag, [312](#)
  - pPdpContext, [312](#)
  - pPdpDataCompType, [312](#)
  - pPdpHdrCompType, [312](#)
  - pPriDNSIPv4AddPref, [312](#)
  - pPriDNSIPv6addpref, [312](#)
  - pPrimaryID, [312](#)
  - pProfileName, [313](#)
  - pProfileNameSize, [313](#)
  - pQosClassID, [313](#)
  - pSecDNSIPv4AddPref, [313](#)
  - pSecDNSIPv6addpref, [313](#)
  - pSecondaryFlag, [313](#)
  - pTFTID1Params, [313](#)
  - pTFTID2Params, [313](#)
  - pUMTSMinQoS, [313](#)
  - pUMTSMinQoSSigInd, [313](#)
  - pUMTSReqQoS, [313](#)
  - pUMTSReqQoSsigInd, [313](#)
  - pUsername, [313](#)
  - pUsernameSize, [313](#)
- LibpackProfile3GPP2, [313](#)
  - pAPNClass3GPP2, [317](#)
  - pAPNEnabled3GPP2, [317](#)
  - pAllowLinger, [317](#)
  - pApnString, [317](#)
  - pApnStringSize, [317](#)
  - pAppPriority, [317](#)
  - pAppType, [317](#)
  - pAuthPassword, [317](#)
  - pAuthPasswordSize, [317](#)
  - pAuthProtocol, [317](#)
  - pAuthRetryCount, [318](#)
  - pAuthTimeout, [318](#)
  - pDataMode, [318](#)
  - pDataRate, [318](#)
  - plpcpAckTimeout, [318](#)
  - plpcpCreqRetryCount, [318](#)
  - plsPcscfAddressNedded, [318](#)
  - pLcpAckTimeout, [318](#)
  - pLcpCreqRetryCount, [318](#)
  - pNegoDnsSrvrPref, [318](#)
  - pPDNInactivTimeout3GPP2, [318](#)
  - pPdnType, [318](#)
  - pPppSessCloseTimer1x, [318](#)
  - pPppSessCloseTimerDO, [318](#)
  - pPriV6DnsAddress, [318](#)
  - pPrimaryV4DnsAddress, [318](#)
  - pRATType, [318](#)
  - pSecV6DnsAddress, [318](#)
  - pSecondaryV4DnsAddress, [318](#)
  - pUserId, [318](#)
  - pUserIdSize, [318](#)
- lineCtrlInfo, [335](#)
  - polarityIncluded, [336](#)
  - pwrDenialTime, [336](#)
  - revPolarity, [336](#)
  - toggleMode, [336](#)
- lineValue
  - voiceALSSelectLineInfo, [981](#)
- linkage
  - altitudeSrcInfo, [100](#)
- list\_type
  - custSettingList, [184](#)
  - DMScustSettingList, [203](#)
  - DMSgetCustomInput, [204](#)
  - getCustomInput, [238](#)
  - pack\_dms\_GetCustFeaturesV2\_t, [521](#)
- listEntries
  - FMSPrefImageList, [225](#)
  - PrefImageList, [598](#)
- listSize
  - FMSImageList, [225](#)
  - FMSPrefImageList, [225](#)
  - ImageList, [286](#)
  - PrefImageList, [598](#)
- loc.h
  - eQMI\_LOC\_SESS\_STATUS\_FAILURE, [1115](#)
  - eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS, [1115](#)
  - eQMI\_LOC\_SESS\_STATUS\_SUCCESS, [1115](#)
  - eQMI\_LOC\_SESS\_STATUS\_TIMEOUT, [1115](#)
- loc.h, [1110](#)
  - LOCEVENTMASKNMEA, [1114](#)
  - LOCEVENTMASKWIFIREQ, [1114](#)
  - pack\_loc\_DeleteAssistData, [1115](#)

- pack\_loc\_EventRegister, 1115
- pack\_loc\_SLQSLOCGetBestAvailPos, 1116
- pack\_loc\_SetExtPowerState, 1115
- pack\_loc\_SetOperationMode, 1116
- pack\_loc\_Start, 1116
- pack\_loc\_Stop, 1117
- unpack\_loc\_BestAvailPos\_Ind, 1117
- unpack\_loc\_DeleteAssistData, 1117
- unpack\_loc\_EngineState\_Ind, 1118
- unpack\_loc\_EventRegister, 1118
- unpack\_loc\_PositionRpt\_Ind, 1118
- unpack\_loc\_SLQSLOCGetBestAvailPos, 1120
- unpack\_loc\_SetExtPowerConfig\_Ind, 1119
- unpack\_loc\_SetExtPowerState, 1119
- unpack\_loc\_SetOperationMode, 1119
- unpack\_loc\_Start, 1120
- unpack\_loc\_Stop, 1120
- loc\_BdsSV, 336
  - id, 336
  - mask, 336
- loc\_BdsSVInfo, 336
  - len, 337
  - pSV, 337
- loc\_CellDb, 337
  - mask, 337
- loc\_ClkInfo, 337
  - mask, 338
- loc\_GnssData, 339
  - mask, 340
- loc\_LocApplicationInfo, 341
  - appNameLength, 342
  - appProviderLength, 342
  - appVersionLength, 342
  - appVersionValid, 342
  - pAppName, 342
  - pAppProvider, 342
  - pAppVersion, 342
- loc\_SV, 344
  - id, 344
  - mask, 344
  - system, 344
- loc\_SVInfo, 344
  - len, 345
  - pSV, 345
- loc\_gpsTime, 340
  - gpsTimeOfWeekMs, 341
  - gpsWeek, 341
- loc\_precisionDilution, 342
  - HDOP, 343
  - PDOP, 343
  - VDOP, 343
- loc\_sensorDataUsage, 343
  - aidingIndicatorMask, 343
  - usageMask, 344
- loc\_svUsedforFix, 345
  - gnssSvUsedList, 346
  - gnssSvUsedList\_len, 346
- LocApplicationInfo, 346
  - appNameLength, 346
  - appProviderLength, 347
  - appVersionLength, 347
  - appVersionValid, 347
  - pAppName, 347
  - pAppProvider, 347
  - pAppVersion, 347
- LocDelAssDataReq, 347
  - pBdsSVInfo, 347
  - pCellDb, 348
  - pClkInfo, 348
  - pGnssData, 348
  - pSVInfo, 348
- LocInjectPositionReq, 350
  - pAltitudeSrcInfo, 353
  - pAltitudeWrtEllipsoid, 353
  - pAltitudeWrtMeanSeaLevel, 353
  - pHorConfidence, 353
  - pHorReliability, 354
  - pHorUncCircular, 354
  - pLatitude, 354
  - pLongitude, 354
  - pPositionSrc, 354
  - pRawHorConfidence, 354
  - pRawHorUncCircular, 354
  - pTimestampAge, 354
  - pTimestampUtc, 354
  - pVertConfidence, 354
  - pVertReliability, 354
  - pVertUnc, 354
- LocInjectSensorDataReq, 354
  - pAcceleroData, 355
  - pAcceleroTempData, 355
  - pAcceleroTimeSrc, 355
  - pGyroData, 355
  - pGyroTempData, 355
  - pGyroTimeSrc, 355
  - pOpaqueIdentifier, 355
- LocSetCradleMountReq, 355
  - pConfidence, 356
  - state, 356
- localTimeOffset
  - nas\_qaQmi3Gpp2TimeZone, 439
  - qaQmi3Gpp2TimeZone, 615
- Location Service(LOC), 51
- logString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 865
- logger
  - common.h, 1071
- longName
  - nasPLMNNameResp, 491
  - PLMNNetworkNameData, 596
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- longNameCI
  - nasPLMNNameResp, 491
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- longNameEn
  - nasPLMNNameResp, 491

- unpack\_nas\_SLQSGetPLMNName\_t, 895
- longNameLen
  - nasPLMNNameResp, 491
  - PLMNNetworkNameData, 596
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- longNameSB
  - nasPLMNNameResp, 491
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- longNameSpareBits
  - PLMNNetworkNameData, 597
- Longitude
  - GPSSStateInfo, 265
- LteBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, 862
- LteCQIParm, 358
  - CQIValueCW0, 359
  - CQIValueCW1, 359
  - ValidityCW0, 359
  - ValidityCW1, 359
- LteEARFCN, 359
  - earfcn0, 359
  - earfcn1, 360
  - status, 360
- LteEmmDI
  - NasSwiIndReg, 497
  - pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 544
- LteEmmUI
  - NasSwiIndReg, 497
  - pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 544
- LteEsmDI
  - NasSwiIndReg, 497
  - pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 544
- LteEsmUI
  - NasSwiIndReg, 497
  - pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 544
- LteGsmCellInfo
  - LTEInfoNeighboringGSM, 366
  - nas\_LTEInfoNeighboringGSM, 423
- LteGsmCellInfo, 360
  - cellReselPriority, 361
  - cells\_len, 361
  - GsmCellInfo, 361
  - nccPermitted, 361
  - threshGsmHigh, 361
  - threshGsmLow, 361
- LteNasReleaseInfo
  - qaGobiApiCb.k.h, 1168
- LteNasReleaseInfo\_s, 367
  - nas\_major, 367
  - nas\_minor, 367
  - nas\_release, 367
- LtePCI, 368
  - earfcn, 368
  - pci, 368
  - status, 368
- LteQci
  - unpack\_qos\_swiQosFlow\_t, 936
- LteRsrpDelta
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- LteRsrpinfo
  - nas\_SLQSSignalStrengthsInformation, 449
  - SLQSSignalStrengthsInformation, 729
- LteRsrpinformation, 368
  - rsrplevel, 369
- LteSSInfo, 375
  - rsrp, 375
  - rsrq, 375
  - rsi, 375
  - snr, 375
- LteSnrDelta
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- LteSnrinfo
  - nas\_SLQSSignalStrengthsInformation, 449
  - SLQSSignalStrengthsInformation, 729
- LteSnrinformation, 372
  - snrlevel, 372
- LteWcdmaCellInfo, 377
  - cellReselPriority, 378
  - cellsLen, 378
  - threshXhigh, 378
  - threshXlow, 378
  - uarfcn, 378
  - WCDMACellInfo, 378
- ltersrp
  - slqsSignalStrengthInfo, 725
  - unpack\_nas\_SLQSGetSignalStrength\_t, 898
- ltesn
  - slqsSignalStrengthInfo, 725
  - unpack\_nas\_SLQSGetSignalStrength\_t, 898
- m\_FwBuildId
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_FwImageld
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_PriBuildId
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_PrImageld
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_nCarrierId
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_nFolderId
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756
- m\_nStorage
  - CarrierImage\_t, 136
  - SWI\_STRUCT\_CarrierImage, 756

- MACIndex
  - NetworkStatEVDO, [507](#)
- MAX\_BUILD\_ID\_LEN
  - dms.h, [1078](#)
  - qaGobiApiDms.h, [1247](#)
- MAX\_CALL\_NO\_LEN
  - qaGobiApiVoice.h, [1485](#)
- MAX\_CONTENT\_LENGTH
  - qaGobiApiUim.h, [1472](#)
- MAX\_CUST\_ID\_LEN
  - qaGobiApiDms.h, [1247](#)
- MAX\_FSN\_LENGTH
  - qaGobiApiDms.h, [1247](#)
- MAX\_ICCID\_LENGTH
  - qaGobiApiUim.h, [1472](#)
  - uim.h, [1584](#)
- MAX\_MSE\_TWS\_MSG
  - sms.h, [1563](#)
- MAX\_NO\_OF\_CALLS
  - qaGobiApiCbk.h, [1165](#)
  - qaGobiApiVoice.h, [1485](#)
- MAX\_NO\_OF\_FILES
  - qaGobiApiCbk.h, [1165](#)
- MAX\_NO\_OF\_SLOTS
  - qaGobiApiCbk.h, [1165](#)
  - qaGobiApiUim.h, [1472](#)
  - uim.h, [1584](#)
- MAX\_NO\_OF\_UUSINFO
  - qaGobiApiCbk.h, [1165](#)
- MAX\_PATH\_LENGTH
  - qaGobiApiCbk.h, [1165](#)
  - qaGobiApiUim.h, [1472](#)
- MAX\_PILOT\_SETS
  - qaGobiApiNas.h, [1321](#)
- MAX\_PUK\_LENGTH
  - qaGobiApiUim.h, [1472](#)
- MAX\_SLOTS\_STATUS
  - qaGobiApiUim.h, [1472](#)
  - uim.h, [1584](#)
- MAX\_SMS\_LIST\_SIZE
  - sms.h, [1563](#)
- MAX\_SMS\_ROUTES
  - qaGobiApiSms.h, [1393](#)
- MAX\_TEMP\_DATA\_LEN
  - qaGobiApiLoc.h, [1311](#)
- MAXUSSDLENGTH
  - qaGobiApiCbk.h, [1165](#)
  - qaGobiApiVoice.h, [1485](#)
- MCC
  - \_SlqsNas3GppNetworkRAT\_, [66](#)
  - CDMASysInfo, [152](#)
  - CDMASysInfoExt, [153](#)
  - currentPLMN, [175](#)
  - GSMSysInfo, [272](#)
  - LTESysInfo, [377](#)
  - nas\_CDMASysInfo, [393](#)
  - nas\_CDMASysInfoExt, [394](#)
  - nas\_currentPLMN, [398](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [429](#)
  - nas\_QmiNas3GppNetworkInfo, [440](#)
  - nas\_QmiNas3GppNetworkRAT, [441](#)
  - nas\_QmisNasPcsDigit, [442](#)
  - nas\_WCDMASysInfo, [467](#)
  - SlqsNas3GppNetworkInfo, [718](#)
  - SlqsNasPcsDigit, [719](#)
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
  - WCDMASysInfo, [1041](#)
- MDMCallDuration
  - ConnectionStatus, [164](#)
  - connectionStatus, [164](#)
- MDMConnStatus
  - ConnectionStatus, [164](#)
  - connectionStatus, [164](#)
- MEIDString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [848](#)
- MIN
  - unpack\_dms\_GetVoiceNumber\_t, [854](#)
- MINREQBKLEN
  - common.h, [1071](#)
- MMTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [939](#)
- MNC
  - \_SlqsNas3GppNetworkRAT\_, [66](#)
  - CDMASysInfo, [152](#)
  - currentPLMN, [175](#)
  - GSMSysInfo, [272](#)
  - LTESysInfo, [377](#)
  - nas\_CDMASysInfo, [393](#)
  - nas\_currentPLMN, [398](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [430](#)
  - nas\_QmiNas3GppNetworkInfo, [440](#)
  - nas\_QmiNas3GppNetworkRAT, [441](#)
  - nas\_QmisNasPcsDigit, [442](#)
  - nas\_WCDMASysInfo, [467](#)
  - SlqsNas3GppNetworkInfo, [718](#)
  - SlqsNasPcsDigit, [719](#)
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
  - WCDMASysInfo, [1041](#)
- MNRInfo, [381](#)
  - mcc, [382](#)
  - mnc, [382](#)
  - rat, [382](#)
- MPTlv
  - NASQmiCbkNasSystemSelPrefInd, [492](#)
- MSGID\_AND\_LEN
  - common.h, [1071](#)
- MSGID\_DONT\_CARE
  - common.h, [1071](#)
- MTMessageInfo
  - newMTMessageTlv, [508](#)
- Mask
  - getDUNCallInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
- mask

- BdsSV, [119](#)
- CellDb, [154](#)
- ClkInfo, [160](#)
- GnssData, [259](#)
- IPv6TrafCls, [306](#)
- loc\_BdsSV, [336](#)
- loc\_CellDb, [337](#)
- loc\_ClkInfo, [338](#)
- loc\_GnssData, [340](#)
- loc\_SV, [344](#)
- SV, [754](#)
- Tos, [789](#)
- unpack\_qos\_IPv6TrafCls\_t, [920](#)
- unpack\_qos\_Tos\_t, [937](#)
- max\_channel\_rx\_rate
  - WDSSWICurrentChannelRates, [1068](#)
- max\_channel\_tx\_rate
  - WDSSWICurrentChannelRates, [1068](#)
- max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [556](#)
  - SwiLocGetAutoStartResp, [758](#)
  - SwiLocSetAutoStartReq, [759](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- max\_dist\_reported
  - SwiLocGetAutoStartResp, [758](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- max\_time
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [556](#)
  - SwiLocGetAutoStartResp, [758](#)
  - SwiLocSetAutoStartReq, [759](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- max\_time\_reported
  - SwiLocGetAutoStartResp, [758](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [943](#)
- MaxAllowedPktSz
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- MaxChanRxRate
  - ChannelRate, [156](#)
  - dunchannelRate, [210](#)
- MaxChanTxRate
  - ChannelRate, [156](#)
  - dunchannelRate, [210](#)
- maxChannelIRXRate
  - unpack\_wds\_GetConnectionRate\_t, [955](#)
- maxChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, [955](#)
- maxDIBitRate
  - LibPackQosClassID, [330](#)
  - QosClassID, [649](#)
- maxDownlinkBitrate
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [838](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- maxImages
  - FMSImageIDEntries, [224](#)
  - ImageIDEntries, [286](#)
- maxMitigationLevel
  - mitigationDevList, [381](#)
- MaxRXChannelRate
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
- maxRxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, [847](#)
- maxSDUSize
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [838](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- maxStorageSize
  - smsMaxStorageSizeResp, [739](#)
- MaxTXChannelRate
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
- maxTxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, [847](#)
- maxUIBitRate
  - LibPackQosClassID, [330](#)
  - QosClassID, [649](#)
- maxUplinkBitrate
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [838](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- mcTimeStamp
  - cdmaMsgDecodingParams, [145](#)
- mcc
  - CSGID, [170](#)
  - MNRInfo, [382](#)
  - nas\_CSGID, [398](#)
  - nas\_MNRInfo, [431](#)
  - nas\_netSelectionPref, [432](#)
  - nasPLMNNameReq, [489](#)
  - netSelectionPref, [502](#)
  - OperatorPLMNData, [520](#)
  - pack\_nas\_SLQSGetPLMNName\_t, [534](#)
  - unpack\_nas\_GetHomeNetwork\_t, [886](#)
- mccM
  - minBasedIMSI, [381](#)
- mccT
  - trueIMSI, [792](#)
- mdmCallDurationActive
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [965](#)
- MdmConnStatus
  - DUNCallInfoInd, [209](#)
- meanThroughputClass
  - GPRSQoS, [261](#)
  - GPRSRequestedQoS, [262](#)
  - LibPackGPRSRequestedQoS, [307](#)
  - wds\_GPRSQoS, [1045](#)
- meid
  - unpack\_dms\_GetSerialNumbers\_t, [853](#)
- meidLength
  - \_SLQSSwiGetSerialNoExtParams, [77](#)
- meidSize
  - serialNumbersInfo, [678](#)
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [848](#)
- message

- unpack\_sms\_SLQSGetSMS\_t, 940
- message\_type
  - NASOTAMessageTlv, 484
- messageClass
  - smsRouteEntry, 744
- messageFailureCode
  - slqssendsmsparams\_s, 722
  - unpack\_sms\_SendSMS\_t, 938
- messageFormat
  - pack\_sms\_SendSMS\_t, 551
  - slqssendasyncsmsparams\_s, 721
  - slqssendsmsparams\_s, 722
  - unpack\_sms\_SLQSGetSMS\_t, 940
- messageID
  - slqssendsmsparams\_s, 722
  - SMSAsyncRawSend\_s, 732
  - unpack\_sms\_SendSMS\_t, 938
- messageId
  - cdmaMsgEncodingParams, 147
- messageIndex
  - pack\_sms\_SLQSGetSMS\_t, 553
  - pack\_sms\_SLQSMModifySMSStatus\_t, 555
  - qmiSmsMessageList, 643
  - SMSMTMessage, 741
  - sSMSMTMessage, 741
- messageLength
  - cdmaMsgDecodingParams, 145
- messageList
  - unpack\_sms\_SLQSGetSMSList\_t, 941
- messageListSize
  - unpack\_sms\_SLQSGetSMSList\_t, 941
- messageMode
  - SMSMemoryInfo, 739
  - SMSMessageMode, 740
  - sSMSMessageMode, 740
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, 941
- MessageModelInfo
  - messageModeTlv, 379
- messageModeTlv, 379
  - MessageModelInfo, 379
  - TlvPresent, 379
- messageSize
  - pack\_sms\_SendSMS\_t, 551
  - slqssendasyncsmsparams\_s, 721
  - slqssendsmsparams\_s, 722
  - unpack\_sms\_SLQSGetSMS\_t, 940
  - wcdmaMsgEncodingParams, 1037
- messageTag
  - pack\_sms\_SLQSMModifySMSStatus\_t, 555
  - qmiSmsMessageList, 643
  - unpack\_sms\_SLQSGetSMS\_t, 940
- messageType
  - smsRouteEntry, 744
- messageWaitingInfoContent, 379
  - activeInd, 380
  - msgCount, 380
  - msgType, 380
- MicMute
  - GetAudioProfileResp, 234
  - GetM2MAudioProfileResp, 251
  - GetM2MAVMuteResp, 253
  - SetAudioProfileReq, 686
  - SetM2MAVMuteReq, 698
- minBasedIMSI, 380
  - imsiM1112, 380
  - imsiMS1, 380
  - imsiMS2, 380
  - mccM, 381
- MinPolicedPktSz
  - unpack\_qos\_swiQosFlow\_t, 936
- minSize
  - unpack\_dms\_GetVoiceNumber\_t, 854
- minute
  - nas\_timeInfo, 456
  - nas\_UniversalTime, 461
  - timeInfo, 784
  - UniversalTime, 843
- mipMode
  - unpack\_wds\_GetMobileIP\_t, 958
- mipStatus
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- mipstatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- mitigationDevID
  - TmdDeRegNotMitigationLvlReq, 785
  - TmdGetMitigationLvlReq, 786
  - TmdMitigationLvlIndReq, 787
  - TmdRegNotMitigationLvlReq, 787
- mitigationDevIDLen
  - TmdDeRegNotMitigationLvlReq, 785
  - TmdGetMitigationLvlReq, 786
  - TmdMitigationLvlIndReq, 787
  - TmdRegNotMitigationLvlReq, 787
- mitigationDevId
  - mitigationDevList, 381
- mitigationDevIdLen
  - mitigationDevList, 381
- MitigationDevInfo
  - QmiCbkTmdMitiLvlRptInd, 639
- mitigationDevList, 381
  - maxMitigationLevel, 381
  - mitigationDevId, 381
  - mitigationDevIdLen, 381
- mnc
  - CSGID, 170
  - MNRInfo, 382
  - nas\_CSGID, 398
  - nas\_MNRInfo, 432
  - nas\_netSelectionPref, 432
  - nasPLMNNNameReq, 489
  - netSelectionPref, 502
  - OperatorPLMNData, 520
  - pack\_nas\_SLQSGetPLMNNName\_t, 534

- unpack\_nas\_GetHomeNetwork\_t, 886
- mncPcsDigits
  - CSGID, 170
  - nas\_CSGID, 398
- mobileCountryCode
  - SMSEtwsPlmn, 736
  - sMSEtwsPlmn, 736
- mobileIP
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 577
- mobileNetworkCode
  - SMSEtwsPlmn, 736
  - sMSEtwsPlmn, 736
- mode
  - callInfo, 132
  - pack\_dms\_SetEventReport\_t, 523
  - pack\_dms\_SetPower\_t, 523
  - pack\_loc\_SetOperationMode\_t, 530
  - UIMRefreshEvent, 823
- ModePref
  - NASModePreferenceTlv, 482
- modelid
  - unpack\_dms\_GetModelID\_t, 851
- modelid\_str
  - slqsfwinfo\_s, 717
  - unpack\_dms\_GetFirmwareInfo\_t, 849
- modemMode
  - CommInfo, 163
  - nas\_CommInfo, 397
- modemTempNotification
  - qaGobiApiCbk.h, 1168
- ModemTempState
  - \_modemTempNotification, 60
- ModemTemperature
  - \_modemTempNotification, 60
- modemindex
  - pack\_fms\_SetImagesPreference\_t, 526
- ModifyProfileIn, 382
  - curProfile, 383
  - pProfileID, 383
  - pProfileType, 383
- ModifyProfileOut, 383
  - pExtErrorCode, 383
- month
  - nas\_timeInfo, 456
  - nas\_UniversalTime, 461
  - timeInfo, 784
  - UniversalTime, 843
- msgCount
  - messageWaitingInfoContent, 380
- msgDelFailureCause
  - SMSAsyncRawSend\_s, 732
- msgDelFailureType
  - SMSAsyncRawSend\_s, 732
- msgProtocol
  - smsMsgprotocolResp, 741
- msgType
  - messageWaitingInfoContent, 380
- msgWaitInfo
  - getMsgWaitingInfo, 255
  - msgWaitingInfo, 384
- msgWaitingInfo, 383
  - msgWaitInfo, 384
  - numInstances, 384
- msgid
  - pack\_qmi\_t, 549
  - unpack\_qmi\_t, 918
- msgtype
  - common.h, 1072
- Mtu
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 967
- MultDisc
  - protocolSubtypeElement, 613
- multiplier
  - pktErrRate, 594
  - unpack\_qos\_pktErrRate\_t, 920
- NAI
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- NAM\_NAME\_LENGTH
  - qaGobiApiNas.h, 1321
- NAS\_PLMN\_LENGTH
  - nas.h, 1126
- NAS\_SRV
  - qaGobiApiCbk.h, 1165
- NASBandPreferenceTlv, 467
  - band\_pref, 467
  - TlvPresent, 467
- NASEmergencyModeTlv, 469
  - EmerMode, 469
  - TlvPresent, 469
- NASGWAcqOrderPrefTlv, 477
  - GWAcqOrderPref, 477
  - TlvPresent, 477
- NASLTEBandPreferenceTlv, 481
  - LTETBandPref, 481
  - TlvPresent, 481
- NASLteNasReleaseInfoTlv, 481
  - nas\_major, 481
  - nas\_minor, 481
  - nas\_release, 481
  - TlvPresent, 481
- NASModePreferenceTlv, 482
  - ModePref, 482
  - TlvPresent, 482
- NASNetSelPreferenceTlv, 482
  - NetSelPref, 482
  - TlvPresent, 482
- NASOTAMessageTlv, 484
  - data\_buf, 484
  - data\_len, 484
  - message\_type, 484
  - TlvPresent, 484
- NASPRLPreferenceTlv, 491
  - PRLPref, 492
  - TlvPresent, 492
- NASPhyCaAggPcellInfo, 484
  - dl\_bw\_value, 485

- freq, [485](#)
- iLTEbandValue, [485](#)
- pci, [485](#)
- TlvPresent, [485](#)
- NASPhyCaAggScellIDBw, [485](#)
  - dl\_bw\_value, [486](#)
  - TlvPresent, [486](#)
- NASPhyCaAggScellIndType, [486](#)
  - freq, [487](#)
  - pci, [487](#)
  - scell\_state, [487](#)
  - TlvPresent, [487](#)
- NASPhyCaAggScellIndex, [486](#)
  - scell\_idx, [486](#)
  - TlvPresent, [486](#)
- NASPhyCaAggScellInfo, [487](#)
  - dl\_bw\_value, [488](#)
  - freq, [488](#)
  - iLTEbandValue, [488](#)
  - pci, [488](#)
  - scell\_state, [488](#)
  - TlvPresent, [488](#)
- NASQmiCbkNasSwiOTAMessageInd, [492](#)
  - nasRelInfoTlv, [492](#)
  - otaMsgTlv, [492](#)
  - timeTlv, [492](#)
- NASQmiCbkNasSystemSelPrefInd, [492](#)
  - BPTlv, [492](#)
  - EMTlv, [492](#)
  - GWAOPTlv, [492](#)
  - LBPTlv, [492](#)
  - MPTlv, [492](#)
  - NSPTlv, [492](#)
  - PRLPTlv, [493](#)
  - RPTlv, [493](#)
  - SDPTlv, [493](#)
- NASRoamPreferenceTlv, [493](#)
  - RoamPref, [493](#)
  - TlvPresent, [493](#)
- NASServDomainPrefTlv, [493](#)
  - SrvDomainPref, [493](#)
  - TlvPresent, [493](#)
- NASServingSystemInfo, [493](#)
  - csAttachState, [494](#)
  - hdrPersonality, [494](#)
  - psAttachState, [494](#)
  - radioInterfaceList, [494](#)
  - radioInterfaceNo, [495](#)
  - registrationState, [495](#)
  - selectedNetwork, [495](#)
- NASTimeInfoTlv, [501](#)
  - time, [501](#)
  - TlvPresent, [501](#)
- NSPTlv
  - NASQmiCbkNasSystemSelPrefInd, [492](#)
- NSSAudioCtrl, [510](#)
  - downLink, [511](#)
  - upLink, [511](#)
- NUM\_OF\_SET
  - qaGobiApiCbk.h, [1165](#)
  - qaGobiApiSms.h, [1393](#)
- NWProfile, [511](#)
  - pProfSz, [511](#)
  - pProfValues, [511](#)
  - tech, [511](#)
- NWQoSStatus
  - unpack\_qos\_SLQSQosGetNetworkStatus\_t, [923](#)
- NWRegStat
  - \_transNWRegInfoNotification, [92](#)
- naiSize
  - unpack\_wds\_GetMobileIPProfile\_t, [959](#)
- namID
  - airTimer, [97](#)
  - nasGet3GPP2SubscriptionInfoReq, [469](#)
  - prefVoiceSO, [600](#)
  - roamTimer, [663](#)
- namName, [384](#)
  - namName, [384](#)
  - namNameLen, [384](#)
  - namName, [384](#)
- namNameLen
  - namName, [384](#)
- Name
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- name
  - unpack\_nas\_GetHomeNetwork\_t, [886](#)
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- nameLen
  - remotePartyName, [658](#)
- namePI
  - remotePartyName, [658](#)
- nameSize
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- namelength
  - omaDmFotaTlv, [516](#)
  - omaDmFotaTlvExt, [518](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- namesize
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- nas.h
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_C-ONFIGURED\_ACTIVATED, [1127](#)
  - eLIBPACK\_NAS\_LTE\_CPHY\_SCELL\_STATE\_C-ONFIGURED\_DEACTIVATED, [1127](#)

- eLIBPACK\_NAS\_LTE\_CPHY\_SELL\_STATE\_DECONFIGURED, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6, [1127](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75, [1127](#)
- eNAS\_LTE\_CPHY\_SELL\_STATE\_CONFIGURED\_ACTIVATED\_LITE, [1127](#)
- eNAS\_LTE\_CPHY\_SELL\_STATE\_CONFIGURED\_DEACTIVATED\_LITE, [1127](#)
- eNAS\_LTE\_CPHY\_SELL\_STATE\_DECONFIGURED\_LITE, [1127](#)
- nas.h, [1121](#)
- NAS\_PLMN\_LENGTH, [1126](#)
- pack\_nas\_GetACCOLC, [1127](#)
- pack\_nas\_GetANAAAuthenticationStatus, [1128](#)
- pack\_nas\_GetCDMANetworkParameters, [1128](#)
- pack\_nas\_GetHomeNetwork, [1128](#)
- pack\_nas\_GetNetworkPreference, [1129](#)
- pack\_nas\_GetRFInfo, [1129](#)
- pack\_nas\_GetServingNetwork, [1129](#)
- pack\_nas\_GetServingNetworkCapabilities, [1129](#)
- pack\_nas\_GetSignalStrengths, [1130](#)
- pack\_nas\_PerformNetworkScan, [1130](#)
- pack\_nas\_SLQSGetNetworkTime, [1132](#)
- pack\_nas\_SLQSGetPLMNName, [1132](#)
- pack\_nas\_SLQSGetServingSystem, [1132](#)
- pack\_nas\_SLQSGetSignalStrength, [1132](#)
- pack\_nas\_SLQSGetSysInfo, [1133](#)
- pack\_nas\_SLQSGetSysSelectionPref, [1133](#)
- pack\_nas\_SLQSIInitiateNetworkRegistration, [1133](#)
- pack\_nas\_SLQSNasConfigSigInfo2, [1134](#)
- pack\_nas\_SLQSNasGetCellLocationInfo, [1134](#)
- pack\_nas\_SLQSNasGetSigInfo, [1134](#)
- pack\_nas\_SLQSNasIndicationRegisterExt, [1135](#)
- pack\_nas\_SLQSNasSwiModemStatus, [1135](#)
- pack\_nas\_SLQSNasSwiOTAMessageCallback, [1135](#)
- pack\_nas\_SLQSSetBandPreference, [1136](#)
- pack\_nas\_SLQSSetSignalStrengthsCallback, [1136](#)
- pack\_nas\_SLQSSetSysSelectionPref, [1136](#)
- pack\_nas\_SLQSSwiGetLteCQI, [1137](#)
- pack\_nas\_SetACCOLC, [1130](#)
- pack\_nas\_SetLURRejectCallback, [1131](#)
- pack\_nas\_SetNetworkPreference, [1131](#)
- pack\_nas\_SetRFInfoCallback, [1131](#)
- pack\_nas\_SlqsGetLTECphyCAInfo, [1131](#)
- unpack\_nas\_GetACCOLC, [1137](#)
- unpack\_nas\_GetANAAAuthenticationStatus, [1137](#)
- unpack\_nas\_GetCDMANetworkParameters, [1138](#)
- unpack\_nas\_GetHomeNetwork, [1138](#)
- unpack\_nas\_GetNetworkPreference, [1138](#)
- unpack\_nas\_GetRFInfo, [1138](#)
- unpack\_nas\_GetServingNetwork, [1139](#)
- unpack\_nas\_GetServingNetworkCapabilities, [1139](#)
- unpack\_nas\_GetSignalStrengths, [1139](#)
- unpack\_nas\_PerformNetworkScan, [1140](#)
- unpack\_nas\_SLQSGetNetworkTime, [1142](#)
- unpack\_nas\_SLQSGetPLMNName, [1142](#)
- unpack\_nas\_SLQSGetServingSystem, [1143](#)
- unpack\_nas\_SLQSGetSignalStrength, [1143](#)
- unpack\_nas\_SLQSGetSysInfo, [1143](#)
- unpack\_nas\_SLQSGetSysSelectionPref, [1144](#)
- unpack\_nas\_SLQSIInitiateNetworkRegistration, [1144](#)
- unpack\_nas\_SLQSNasConfigSigInfo2, [1144](#)
- unpack\_nas\_SLQSNasGetCellLocationInfo, [1144](#)
- unpack\_nas\_SLQSNasGetSigInfo, [1145](#)
- unpack\_nas\_SLQSNasIndicationRegisterExt, [1145](#)
- unpack\_nas\_SLQSNasNetworkTimeCallback\_ind, [1145](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind, [1146](#)
- unpack\_nas\_SLQSNasSwiModemStatus, [1146](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback, [1146](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind, [1147](#)
- unpack\_nas\_SLQSNasSysInfoCallback\_ind, [1147](#)
- unpack\_nas\_SLQSSetBandPreference, [1147](#)
- unpack\_nas\_SLQSSetSignalStrengthsCallback, [1147](#)
- unpack\_nas\_SLQSSetSysSelectionPref, [1148](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallback\_ind, [1148](#)
- unpack\_nas\_SLQSSwiGetLteCQI, [1148](#)
- unpack\_nas\_SetACCOLC, [1140](#)
- unpack\_nas\_SetDataCapabilitiesCallback\_ind, [1140](#)
- unpack\_nas\_SetEventReportInd, [1141](#)
- unpack\_nas\_SetLURRejectCallback, [1141](#)
- unpack\_nas\_SetNasLTECphyCAIndCallback\_ind, [1141](#)
- unpack\_nas\_SetNetworkPreference, [1141](#)
- unpack\_nas\_SetRFInfoCallback, [1141](#)
- unpack\_nas\_SetRoamingIndicatorCallback\_ind, [1142](#)
- unpack\_nas\_SetServingSystemCallback\_ind, [1142](#)
- unpack\_nas\_SlqsGetLTECphyCAInfo, [1142](#)
- nas\_AddCDMASysInfo, [385](#)
- geoSysIdx, [385](#)
- regPrd, [385](#)
- nas\_AddSysInfo, [385](#)
- cellBroadcastCap, [386](#)
- geoSysIdx, [386](#)
- nas\_CDMAECIOThresh, [388](#)
- CDMAECIOThreshListLen, [388](#)
- pCDMAECIOThreshList, [388](#)
- nas\_CDMAInfo, [388](#)

- baseId, [389](#)
- baseLat, [389](#)
- baseLong, [389](#)
- nid, [389](#)
- refpn, [389](#)
- sid, [389](#)
- nas\_CDMARSSIThresh, [389](#)
  - CDMARSSIThreshListLen, [390](#)
  - pCDMARSSIThreshList, [390](#)
- nas\_CDMASysInfo, [390](#)
  - baseId, [393](#)
  - baseLat, [393](#)
  - baseLong, [393](#)
  - bsInfoValid, [393](#)
  - bsPRev, [393](#)
  - bsPRevValid, [393](#)
  - ccsSupported, [393](#)
  - ccsSupportedValid, [393](#)
  - cdmaSysIdValid, [393](#)
  - isSysPrIMatch, [393](#)
  - isSysPrIMatchValid, [393](#)
  - MCC, [393](#)
  - MNC, [393](#)
  - networkID, [393](#)
  - networkIdValid, [393](#)
  - pRevInUse, [393](#)
  - pRevInUseValid, [393](#)
  - packetZone, [393](#)
  - packetZoneValid, [393](#)
  - sysInfoCDMA, [393](#)
  - systemID, [393](#)
- nas\_CDMASysInfoExt, [393](#)
  - imsi\_11\_12, [394](#)
  - MCC, [394](#)
- nas\_CSGID, [397](#)
  - id, [397](#)
  - mcc, [398](#)
  - mnc, [398](#)
  - mncPcsDigits, [398](#)
  - rat, [398](#)
- nas\_CallBarringSysInfo, [386](#)
  - csBarStatus, [387](#)
  - psBarStatus, [387](#)
- nas\_CommInfo, [395](#)
  - imsRegState, [397](#)
  - modemMode, [397](#)
  - psState, [397](#)
  - systemMode, [397](#)
  - temperature, [397](#)
- nas\_GERANInfo, [402](#)
  - arfcn, [404](#)
  - bsic, [404](#)
  - cellID, [404](#)
  - insNmrCellInfo, [404](#)
  - lac, [404](#)
  - nmrInst, [404](#)
  - plmn, [404](#)
  - rxLev, [404](#)
  - timingAdvance, [404](#)
- nas\_GSMRSSIThresh, [406](#)
  - GSMRSSIThreshListLen, [407](#)
  - pGSMRSSIThreshList, [407](#)
- nas\_GSMSrvStatusInfo, [407](#)
  - isPrefDataPath, [408](#)
  - srvStatus, [408](#)
  - trueSrvStatus, [408](#)
- nas\_GSMSysInfo, [408](#)
  - cellId, [410](#)
  - cellIdValid, [410](#)
  - dtmSupp, [410](#)
  - dtmSuppValid, [410](#)
  - egprsSupp, [410](#)
  - egprsSuppValid, [410](#)
  - lac, [410](#)
  - lacValid, [410](#)
  - MCC, [410](#)
  - MNC, [410](#)
  - networkIdValid, [410](#)
  - regRejectInfoValid, [410](#)
  - rejCause, [410](#)
  - rejectSrvDomain, [411](#)
  - sysInfoGSM, [411](#)
- nas\_HDRECIOThresh, [411](#)
  - HDRECIOThreshListLen, [411](#)
  - pHDRECIOThreshList, [411](#)
- nas\_HDRIOThresh, [411](#)
  - HDRIOThreshListLen, [412](#)
  - pHDRIOThreshList, [412](#)
- nas\_HDRRSSIThresh, [412](#)
  - HDRRSSIThreshListLen, [412](#)
  - pHRRSSIThreshList, [412](#)
- nas\_HDRSINRThreshold, [412](#)
  - HDRSINRThresholdListLen, [413](#)
  - pHDRSINRThresholdList, [413](#)
- nas\_HDRSysInfo, [413](#)
  - hdrActiveProt, [415](#)
  - hdrActiveProtValid, [415](#)
  - hdrPersonality, [415](#)
  - hdrPersonalityValid, [415](#)
  - is856SysId, [415](#)
  - is856SysIdValid, [415](#)
  - isSysPrIMatch, [415](#)
  - isSysPrIMatchValid, [415](#)
  - sysInfoHDR, [415](#)
- nas\_LTEInfo, [417](#)
  - band, [419](#)
  - bandwidth, [419](#)
  - emmConnState, [419](#)
  - emmState, [419](#)
  - emmSubState, [419](#)
  - RXChan, [419](#)
  - TXChan, [419](#)
- nas\_LTEInfoInterfreq, [420](#)
  - freqsLen, [420](#)
  - InfoInterfreq, [420](#)
  - ueInIdle, [420](#)

- nas\_LTEInfoIntrafreq, 420
  - CellParams, 422
  - cellReselPriority, 422
  - cellsLen, 422
  - earfcn, 422
  - globalCellId, 422
  - plmn, 422
  - sIntraSearch, 422
  - sNonIntraSearch, 422
  - servingCellId, 422
  - tac, 422
  - threshServingLow, 422
  - ueInIdle, 422
- nas\_LTEInfoNeighboringGSM, 423
  - freqsLen, 423
  - LteGsmCellInfo, 423
  - ueInIdle, 423
- nas\_LTEInfoNeighboringWCDMA, 423
  - freqsLen, 424
  - ueInIdle, 424
- nas\_LTERSRPThresh, 424
  - LTERSRPThreshListLen, 424
  - pLTERSRPThreshList, 425
- nas\_LTERSRQThresh, 425
  - LTERSRQThreshListLen, 425
  - pLTERSRQThreshList, 425
- nas\_LTERSSIThresh, 425
  - LTERSSIThreshListLen, 426
  - pLTERSSIThreshList, 426
- nas\_LTESNRThreshold, 427
  - LTESNRThresholdListLen, 427
  - pLTESNRThresholdList, 427
- nas\_LTESigRptConfig, 426
  - avgPeriod, 426
  - rptRate, 426
- nas\_LTESysInfo, 427
  - cellId, 429
  - cellIdValid, 429
  - lac, 429
  - lacValid, 429
  - MCC, 429
  - MNC, 430
  - networkIdValid, 430
  - regRejectInfoValid, 430
  - rejCause, 430
  - rejectSrvDomain, 430
  - sysInfoLTE, 430
  - tac, 430
  - tacValid, 430
- nas\_MNRInfo, 431
  - mcc, 431
  - mnc, 432
  - rat, 432
- nas\_PhyCaAggPcellInfo, 434
  - dl\_bw\_value, 434
  - freq, 434
  - iLTEbandValue, 434
  - pci, 434
  - TlvPresent, 434
- nas\_PhyCaAggScellIDBw, 434
  - dl\_bw\_value, 435
  - TlvPresent, 435
- nas\_PhyCaAggScellIndType, 435
  - freq, 436
  - pci, 436
  - scell\_state, 436
  - TlvPresent, 436
- nas\_PhyCaAggScellIndex, 435
  - scell\_idx, 435
  - TlvPresent, 435
- nas\_PhyCaAggScellInfo, 436
  - dl\_bw\_value, 439
  - freq, 439
  - iLTEbandValue, 439
  - pci, 439
  - scell\_state, 439
  - TlvPresent, 439
- nas\_QmiNas3GppNetworkInfo, 440
  - Description, 440
  - Forbidden, 440
  - InUse, 440
  - MCC, 440
  - MNC, 440
  - Preferred, 440
  - Roaming, 440
- nas\_QmiNas3GppNetworkRAT, 440
  - MCC, 441
  - MNC, 441
  - RAT, 441
- nas\_QmisNasPcsDigit, 441
  - includes\_pcs\_digit, 441
  - MCC, 442
  - MNC, 442
- nas\_RFInfoTlv, 442
  - activeBandClass, 442
  - activeChannel, 442
  - radioInterface, 443
  - radioInterfaceSize, 443
  - TlvPresent, 443
- nas\_RejectReasonTlv, 442
  - rejectCause, 442
  - serviceDomain, 442
  - TlvPresent, 442
- nas\_SLQSSignalStrengthsIndReq, 447
  - ecioDelta, 448
  - ecioThresholdList, 448
  - ecioThresholdListLen, 448
  - ioDelta, 448
  - lteRsrpDelta, 448
  - lteSnrDelta, 448
  - rsrqDelta, 448
  - rxSignalStrengthDelta, 448
  - sinrDelta, 448
  - sinrThresholdList, 448
  - sinrThresholdListLen, 448
- nas\_SLQSSignalStrengthsInformation, 448

- ecioInfo, 449
- errorRateInfo, 449
- io, 449
- lteRsrpinfo, 449
- lteSnrinfo, 449
- rsrqInfo, 449
- rxSignalStrengthInfo, 449
- sinr, 449
- nas\_SLQSSignalStrengthsTlv, 449
  - sSLQSSignalStrengthsInfo, 449
  - TlvPresent, 449
- nas\_SignalStrengthTlv, 447
  - radiolInterface, 447
  - signalStrength, 447
  - TlvPresent, 447
- nas\_SrvStatusInfo, 449
  - isPrefDataPath, 450
  - srvStatus, 450
- nas\_TDSCDMAECIOThresh, 453
- nas\_TDSCDMARSCPTThresh, 453
- nas\_TDSCDMARSSIThresh, 454
- nas\_TDSCDMASINRThresh, 454
- nas\_UMTSInfo, 456
  - cellID, 458
  - ecio, 458
  - geranInst, 458
  - GeranInstInfo, 458
  - lac, 458
  - plmn, 458
  - psc, 458
  - rscp, 458
  - UMTSInstInfo, 458
  - uarfcn, 458
  - umtsInst, 458
- nas\_UMTSinstInfo, 458
  - umtsEcio, 459
  - umtsPsc, 459
  - umtsRscp, 459
  - umtsUarfcn, 459
- nas\_UniversalTime, 460
  - day, 461
  - dayOfWeek, 461
  - hour, 461
  - minute, 461
  - month, 461
  - second, 461
  - year, 461
- nas\_WCDMAECIOThresh, 462
- nas\_WCDMAInfoLTENeighborCell, 463
  - umtsLTENbrCellLen, 463
  - wcdmaRRCState, 463
- nas\_WCDMARSSIThresh, 463
- nas\_WCDMASysInfo, 464
  - cellId, 467
  - cellIdValid, 467
  - hsCallStatus, 467
  - hsCallStatusValid, 467
  - hsInd, 467
  - hsIndValid, 467
  - lac, 467
  - lacValid, 467
  - MCC, 467
  - MNC, 467
  - networkIdValid, 467
  - psc, 467
  - pscValid, 467
  - regRejectInfoValid, 467
  - rejCause, 467
  - rejectSrvDomain, 467
  - sysInfoWCDMA, 467
- nas\_acqOrderPref, 384
  - acqOrdeLen, 385
  - pAcqOrder, 385
- nas\_callBarStatus, 387
  - csBarStatus, 388
  - psBarStatus, 388
- nas\_cellParams, 394
  - pci, 395
  - rsrp, 395
  - rsrq, 395
  - rsi, 395
  - srxlev, 395
- nas\_currentPLMN, 398
  - MCC, 398
  - MNC, 398
  - netDescr, 398
  - netDescrLength, 399
- nas\_dataSrvCapabilities, 399
  - dataCapabilities, 399
  - dataCapabilitiesLen, 399
- nas\_detailSvcInfo, 399
  - hdrHybrid, 401
  - hdrSrvStatus, 401
  - isSysForbidden, 401
  - srvCapability, 401
  - srvStatus, 401
- nas\_ecioListElement, 401
  - ecio, 401
  - radiolf, 401
- nas\_errorRateListElement, 402
  - errorRate, 402
  - radiolf, 402
- nas\_geranInstInfo, 404
  - geranArfcn, 405
  - geranBsicBcc, 405
  - geranBsicNcc, 405
  - geranRssi, 405
- nas\_gsmCellInfo, 405
  - arfcn, 406
  - band1900, 406
  - bsicId, 406
  - cellIdValid, 406
  - rssi, 406
  - srxlev, 406
- nas\_infoInterFreq, 415
  - cell\_resel\_priority, 416

- cellInterFreqParams, 416
- cells\_len, 416
- earfcn, 416
- threshXHigh, 416
- threshXLow, 416
- nas\_lteGsmCellInfo, 416
  - cellReselPriority, 417
  - cells\_len, 417
  - GsmCellInfo, 417
  - nccPermitted, 417
  - threshGsmHigh, 417
  - threshGsmLow, 417
- nas\_lteRsrpInformation, 424
  - rsrplevel, 424
- nas\_lteSnrinformation, 426
  - snrlevel, 427
- nas\_lteWcdmaCellInfo, 430
  - cellReselPriority, 431
  - cellsLen, 431
  - threshXhigh, 431
  - threshXlow, 431
  - uarfcn, 431
  - WCDMACellInfo, 431
- nas\_major
  - LteNasReleaseInfo\_s, 367
  - NASLteNasReleaseInfoTlv, 481
- nas\_minor
  - LteNasReleaseInfo\_s, 367
  - NASLteNasReleaseInfoTlv, 481
- nas\_netSelectionPref, 432
  - mcc, 432
  - mnc, 432
  - netReg, 432
- nas\_nmrCellInfo, 432
  - nmrArfcn, 433
  - nmrBsic, 433
  - nmrCellID, 433
  - nmrLac, 433
  - nmrPlmn, 433
  - nmrRxLev, 433
- nas\_qaQmi3Gpp2TimeZone, 439
  - daylightSavings, 439
  - leapSeconds, 439
  - localTimeOffset, 439
- nas\_release
  - LteNasReleaseInfo\_s, 367
  - NASLteNasReleaseInfoTlv, 481
- nas\_roamIndList, 443
  - numInstances, 443
  - radioInterface, 443
  - roamIndicator, 443
- nas\_rsrqInformation, 444
  - radioIf, 444
  - rsrq, 444
- nas\_rxSignalStrengthListElement, 444
  - radioIf, 445
  - rxSignalStrength, 445
- nas\_servSystem, 445
- csAttachState, 446
- numRadioInterfaces, 446
- psAttachState, 446
- radioInterface, 446
- regState, 446
- selNetwork, 446
- nas\_sysInfoCommon, 450
  - isSysForbidden, 452
  - isSysForbiddenValid, 452
  - roamStatus, 452
  - roamStatusValid, 452
  - srvCapability, 452
  - srvCapabilityValid, 452
  - srvDomain, 452
  - srvDomainValid, 452
- nas\_timeInfo, 455
  - day, 456
  - dayLtSavingAdj, 456
  - dayOfWeek, 456
  - hour, 456
  - minute, 456
  - month, 456
  - radioInterface, 456
  - second, 456
  - timeZone, 456
  - TlvPresent, 456
  - year, 456
- nas\_umtsLTENbrCell, 459
  - cellsTDD, 460
  - earfcn, 460
  - pci, 460
  - rsrp, 460
  - rsrq, 460
  - srxlev, 460
- nas\_wcdmaCellInfo, 461
  - cpich\_ecno, 462
  - cpich\_rscp, 462
  - psc, 462
  - srxlev, 462
- nasCellLocationInfoResp, 468
  - pCDMAInfo, 468
  - pGERANInfo, 469
  - pLTEInfoInterfreq, 469
  - pLTEInfoIntrafreq, 469
  - pLTEInfoNeighboringGSM, 469
  - pLTEInfoNeighboringWCDMA, 469
  - pUMTSCellID, 469
  - pUMTSInfo, 469
  - pWCDMAInfoLTENeighborCell, 469
- nasGet3GPP2SubscriptionInfoReq, 469
  - namID, 469
- nasGet3GPP2SubscriptionInfoResp, 470
  - pCDMAChannel, 470
  - pDirNum, 470
  - pHomeSIDNID, 470
  - pMinBasedIMSI, 470
  - pNAMNameInfo, 470
  - pTrueIMSI, 470

- nasGetHDRColorCodeResp, 470
  - pColorCode, 471
- nasGetLTECphyCa, 471
  - sPhyCaAggPcellInfo, 471
  - sPhyCaAggScellIDBw, 471
  - sPhyCaAggScellIndType, 471
  - sPhyCaAggScellIndex, 471
  - sPhyCaAggScellInfo, 471
- NasGetLTECphyCaInfo, 471
  - PhyCaAggPcellInfo, 472
  - PhyCaAggScellIDBw, 472
  - PhyCaAggScellIndType, 472
  - PhyCaAggScellIndex, 472
  - PhyCaAggScellInfo, 472
- nasGetLTECphyCaResp, 472
  - pPhyCaAggPcellInfo, 472
  - pPhyCaAggScellIDBw, 472
  - pPhyCaAggScellIndType, 472
  - pPhyCaAggScellIndex, 472
  - pPhyCaAggScellInfo, 472
- nasGetSigInfoResp, 472
  - pCDMASSInfo, 473
  - pGSMSSInfo, 473
  - pHDRSSInfo, 473
  - pLTESSInfo, 473
  - pTDSCDMASigInfoExt, 473
  - pTDSCDMASigInfoRscp, 473
  - pWCDMASSInfo, 473
- nasGetSysInfoResp, 473
  - pAddCDMASysInfo, 475
  - pAddGSMSSysInfo, 475
  - pAddHDRSysInfo, 475
  - pAddLTESysInfo, 475
  - pAddWCDMASysInfo, 475
  - pCDMASrvStatusInfo, 475
  - pCDMASysInfo, 475
  - pGSMCallBarringSysInfo, 475
  - pGSMCipherDomainSysInfo, 476
  - pGSMSSrvStatusInfo, 476
  - pGSMSSysInfo, 476
  - pHDRSrvStatusInfo, 476
  - pHDRSysInfo, 476
  - pLTESrvStatusInfo, 476
  - pLTESysInfo, 476
  - pLTEVoiceSupportSysInfo, 476
  - pWCDMACallBarringSysInfo, 476
  - pWCDMACipherDomainSysInfo, 476
  - pWCDMASrvStatusInfo, 476
  - pWCDMASysInfo, 476
- nasGetTxRxInfoReq, 476
  - radio\_if, 476
- nasGetTxRxInfoResp, 477
  - pRXChain0Info, 477
  - pRXChain1Info, 477
  - pTXInfo, 477
- nasIndicationRegisterReq, 477
  - pDDTMInd, 480
  - pDualStandByPrefInd, 480
  - pErrorRateInd, 480
  - pHDRNewUATIAssInd, 480
  - pHDRSessionCloseInd, 480
  - pLTECphyCa, 480
  - pManagedRoamingInd, 480
  - pNetworkTimeInd, 480
  - pServingSystemInd, 480
  - pSignalStrengthInd, 480
  - pSubscriptionInfoInd, 480
  - pSysInfoInd, 480
  - pSystemSelectionInd, 480
- nasInitNetworkReg, 480
  - pChangeDuration, 481
  - pMNRInfo, 481
  - pMncPcsDigitStatus, 481
  - regAction, 481
- nasNetworkTime, 482
  - pDayltSavAdj, 483
  - pRadioInterface, 483
  - pTimeZone, 483
  - universalTime, 483
- nasOperatorNameResp, 483
  - pNITZInformation, 484
  - pOperatorNameString, 484
  - pOperatorPLMNList, 484
  - pPLMNNetworkName, 484
  - pSrvProviderName, 484
- nasPLMNNameReq, 488
  - mcc, 489
  - mnc, 489
  - pMncPcsStatus, 489
- nasPLMNNameResp, 489
  - longName, 491
  - longNameCI, 491
  - longNameEn, 491
  - longNameLen, 491
  - longNameSB, 491
  - shortName, 491
  - shortNameCI, 491
  - shortNameEn, 491
  - shortNameLen, 491
  - shortNameSB, 491
  - spn, 491
  - spnEncoding, 491
  - spnLength, 491
- nasRelInfoTlv
  - NASQmiCbkNasSwiOTAMessageInd, 492
- nasSigInfo, 495
  - pCDMASigInfo, 495
  - pGSMSigInfo, 495
  - pHDRSigInfo, 496
  - pLTESigInfo, 496
  - pRscp, 496
  - pTDSCDMASigInfoExt, 496
  - pWCDMASigInfo, 496
- nasSwiGetChannelLockResp, 496
  - pLteEARFCN, 496
  - pLtePCI, 496

- pWcdmaUARFCN, 496
- NasSwiIndReg, 496
  - gsmUmtsDI, 497
  - gsmUmtsUI, 497
  - lteEmmDI, 497
  - lteEmmUI, 497
  - lteEsmDI, 497
  - lteEsmUI, 497
  - pRankIndicatorInd, 497
- nasSwiSetChannelLockReq, 498
  - pLteEARFCN, 498
  - pLtePCI, 498
  - pWcdmaUARFCN, 498
- nasSysInfo, 498
  - pAddCDMASysInfo, 500
  - pAddGSMSysInfo, 500
  - pAddHDRSysInfo, 500
  - pAddLTESysInfo, 500
  - pAddWCDMASysInfo, 500
  - pCDMASrvStatusInfo, 500
  - pCDMASysInfo, 500
  - pGSMCallBarringSysInfo, 500
  - pGSMCipherDomainSysInfo, 501
  - pGSMSrvStatusInfo, 501
  - pGSMSysInfo, 501
  - pHDRSrvStatusInfo, 501
  - pHDRSysInfo, 501
  - pLTESrvStatusInfo, 501
  - pLTESysInfo, 501
  - pLTEVoiceSupportSysInfo, 501
  - pSysInfoNoChange, 501
  - pWCDMACallBarringSysInfo, 501
  - pWCDMACipherDomainSysInfo, 501
  - pWCDMASrvStatusInfo, 501
  - pWCDMASysInfo, 501
- nccPermitted
  - lteGsmCellInfo, 361
  - nas\_lteGsmCellInfo, 417
- NeighborSetCnt
  - NetworkStat1x, 506
- netDescr
  - currentPLMN, 175
  - nas\_currentPLMN, 398
- netDescrLength
  - currentPLMN, 175
  - nas\_currentPLMN, 399
- netInfoLen
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- netReg
  - nas\_netSelectionPref, 432
  - netSelectionPref, 502
- NetSelPref
  - NASNetSelPreferenceTlv, 482
- netSelectionPref, 501
  - mcc, 502
  - mnc, 502
  - netReg, 502
- NetStats, 502
  - rx\_bytes, 503
  - rx\_errors, 503
  - rx\_overflows, 503
  - rx\_packets, 503
  - tx\_bytes, 503
  - tx\_errors, 503
  - tx\_overflows, 503
  - tx\_packets, 503
- Network Access Service (NAS), 34
- NetworkDebugResp, 503
  - pDataStatusDetail, 504
  - pDeviceConfigDetail, 504
  - pNetworkStat1x, 504
  - pNetworkStatEVDO, 504
  - pObjectVer, 504
- NetworkID
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- networkID
  - CDMASysInfo, 152
  - nas\_CDMASysInfo, 393
- networkIdValid
  - CDMASysInfo, 152
  - GSMSysInfo, 272
  - LTESysInfo, 377
  - nas\_CDMASysInfo, 393
  - nas\_GSMSysInfo, 410
  - nas\_LTESysInfo, 430
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- networkInfoLen
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 963
- NetworkStat1x, 504
  - ActSetCnt, 506
  - NeighborSetCnt, 506
  - pActPilotPNElements, 506
  - pNeighborSetPilotPN, 506
  - RX\_EC\_IO, 506
  - RX\_PWR, 506
  - SO, 506
  - State, 506
  - TX\_PWR, 506
- NetworkStatEVDO, 506
  - MACIndex, 507
  - PER, 508
  - pSectorID, 508
  - PilotEnergy, 508
  - RX\_PWR, 508
  - SNR, 508
  - SectorIDLen, 508
  - State, 508
- NetworkType
  - CurrNetworkInfo, 178
  - currNetworkInfo, 178
  - wds\_currNetworkInfo, 1043
- NewMMTlv

- unpack\_sms\_SetNewSMSCallback\_ind\_t, 939
- newMTMessageTlv, 508
  - MTMessageInfo, 508
  - TlvPresent, 508
- newPINLen
  - uim\_unblockUIMPIN, 809
  - unblockUIMPIN, 842
- newPINVal
  - uim\_unblockUIMPIN, 809
  - unblockUIMPIN, 842
- newPasswd
  - voiceSetCallBarringPwdInfo, 1018
- newPasswdAgain
  - voiceSetCallBarringPwdInfo, 1018
- newPwd
  - newPwdData, 509
- newPwdAgain
  - newPwdData, 509
- newPwdData, 508
  - newPwd, 509
  - newPwdAgain, 509
- nextHeader
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- nid
  - CDMAInfo, 143
  - nas\_CDMAInfo, 389
  - sidNid, 708
  - unpack\_nas\_GetHomeNetwork\_t, 886
- NmeaPort
  - DcsUsbPortNames, 193
- nmrArfcn
  - nas\_nmrCellInfo, 433
  - nmrCellInfo, 510
- nmrBsic
  - nas\_nmrCellInfo, 433
  - nmrCellInfo, 510
- nmrCellID
  - nas\_nmrCellInfo, 433
  - nmrCellInfo, 510
- nmrCellInfo, 509
  - nmrArfcn, 510
  - nmrBsic, 510
  - nmrCellID, 510
  - nmrLac, 510
  - nmrPlmn, 510
  - nmrRxLev, 510
- nmrInst
  - GERANInfo, 228
  - nas\_GERANInfo, 404
- nmrLac
  - nas\_nmrCellInfo, 433
  - nmrCellInfo, 510
- nmrPlmn
  - nas\_nmrCellInfo, 433
  - nmrCellInfo, 510
- nmrRxLev
  - nas\_nmrCellInfo, 433
- nmrCellInfo, 510
- noReplyTimer
  - callFWExtInfo, 129
  - callFWInfo, 130
- Non-service specific APIs (SWI), 45
- notifType
  - voiceSUPSNotification, 1031
- notification
  - omaDmNotificationsTlv, 518
  - unpack\_omaDmNotificationsTlv\_t, 917
- notificationType
  - SMSEtwSMessage, 735
  - sMSEtwSMessage, 734
- notused
  - unpack\_dms\_SetCrashAction\_t, 854
- num\_instances
  - \_qaQmi3GPP2BroadcastCfgInfo, 63
  - \_qaQmi3GPPBroadcastCfgInfo, 63
  - custSettingList, 184
  - DMScustSettingList, 203
- numApp
  - slotInf, 712
  - slotInfo, 714
  - uim\_slotInfo, 807
- numCrashes
  - CrashInfo, 167
- numEntries
  - CurrentImgList, 174
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 864
- numFeatures
  - personalizationStatus, 588
- numFiles
  - registerRefresh, 657
- NumFlows
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, 926
- numInstance
  - operatorPLMNList, 520
  - PLMNNetworkName, 595
- numInstances
  - arrAlertingPattern, 107
  - arrAlertingType, 108
  - arrAlphaID, 108
  - arrCalledPartyNum, 109
  - arrCallEndReason, 110
  - arrCallInfo, 110
  - arrConnectPartyNum, 111
  - arrDiagInfo, 111
  - arrRedirPartyNum, 112
  - arrRemotePartyName, 112
  - arrRemotePartyNum, 113
  - arrSvcOption, 113
  - arrUUSInfo, 114
  - DomainNameList, 207
  - getCallFWExtInfo, 236
  - getCallFWInfo, 237
  - getMsgWaitingInfo, 255
  - homeSIDNID, 282

- msgWaitingInfo, [384](#)
- nas\_roamIndList, [443](#)
- PCSCFFQDNAddressList, [582](#)
- PCSCFIPv4ServerAddressList, [582](#)
- roamIndList, [662](#)
- wds\_DomainNameList, [1044](#)
- wds\_PCSCFFQDNAddressList, [1047](#)
- wds\_PCSCFIPv4ServerAddressList, [1047](#)
- numLen
  - calledPartyInfo, [124](#)
  - callFWExtInfo, [129](#)
  - callFWInfo, [130](#)
  - callingPartyInfo, [133](#)
  - peerNumberInfo, [587](#)
  - redirNumInfo, [656](#)
  - remotePartyNum, [659](#)
- numOfFiles
  - UIMRefreshEvent, [823](#)
- numOfRoutes
  - smsSetRoutesReq, [744](#)
- numOpt
  - DHCPOptionList, [199](#)
  - WdsDHCPv4OptionList, [1059](#)
  - wdsDhcpv4OptionList, [1058](#)
- numPI
  - peerNumberInfo, [587](#)
- NumPilots
  - PilotSetData, [593](#)
- numPlan
  - calledPartyInfo, [124](#)
  - callFWExtInfo, [129](#)
  - callingPartyInfo, [133](#)
  - connectNumInfo, [166](#)
  - peerNumberInfo, [587](#)
  - redirNumInfo, [656](#)
- numPresInd
  - connectNumInfo, [166](#)
- numQosFlow
  - sQosStat, [748](#)
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [926](#)
- numRadioInterfaces
  - nas\_servSystem, [446](#)
  - servSystem, [681](#)
- NumRxFilters
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- numSI
  - peerNumberInfo, [587](#)
- numSlot
  - cardStatus, [135](#)
  - uim\_cardStatus, [800](#)
- NumSupUSBComps
  - unpack\_dms\_GetUSBComp\_t, [854](#)
- NumTxFilters
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- numType
  - calledPartyInfo, [124](#)
  - callFWExtInfo, [129](#)
  - callingPartyInfo, [133](#)
  - connectNumInfo, [166](#)
  - peerNumberInfo, [587](#)
  - redirNumInfo, [656](#)
- number
  - calledPartyInfo, [124](#)
  - callFWExtInfo, [129](#)
  - callFWInfo, [130](#)
  - callingPartyInfo, [133](#)
  - ECTNum, [212](#)
  - peerNumberInfo, [587](#)
  - redirNumInfo, [656](#)
- numberPlan
  - callFwdTypeAndPlan, [127](#)
- numberType
  - callFwdTypeAndPlan, [127](#)
- NxtHdrProto
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- OKtoRefresh
  - UIMRefreshOKReq, [825](#)
- OMADMCancelSession
  - qaGobiApiOadm.h, [1363](#)
- OMADMEnabled
  - unpack\_swima\_SLQSOMADMGetSettings\_t, [948](#)
- OMADMGetPendingNIA
  - qaGobiApiOadm.h, [1363](#)
- OMADMGetSessionInfo
  - qaGobiApiOadm.h, [1364](#)
- OMADMStartSession
  - qaGobiApiOadm.h, [1365](#)
- OTASPStatus
  - voiceOTASPStatusInfo, [1014](#)
- oddEvenInd
  - calledPartySubAdd, [125](#)
- OfflineReason
  - unpack\_dms\_GetPower\_t, [852](#)
- offset
  - readTransparentInfo, [654](#)
  - uim\_readTransparentInfo, [803](#)
- oldPINLen
  - changeUIMPIN, [156](#)
  - uim\_changeUIMPIN, [800](#)
- oldPINVal
  - changeUIMPIN, [156](#)
  - uim\_changeUIMPIN, [801](#)
- oldPasswd
  - voiceSetCallBarringPwdInfo, [1018](#)
- omaDmConfig
  - sessionInfo, [682](#)
  - sessionInfoExt, [682](#)
- omaDmConfigTlv, [511](#)
  - alertmsg, [512](#)
  - alertmsglength, [512](#)
  - state, [512](#)
  - userInputReq, [512](#)
  - userInputTimeout, [512](#)
- omaDmConfigTlvExt, [512](#)
  - alertmsg, [514](#)
  - alertmsglength, [514](#)

- state, [514](#)
- userInputReq, [514](#)
- userInputTimeout, [514](#)
- omaDmFota
  - sessionInfo, [682](#)
  - sessionInfoExt, [682](#)
- omaDmFotaTlv, [514](#)
  - description, [515](#)
  - descriptionlength, [515](#)
  - fwloadsize, [515](#)
  - fwloadComplete, [515](#)
  - namelength, [516](#)
  - package\_name, [516](#)
  - sessionType, [516](#)
  - severity, [516](#)
  - state, [516](#)
  - updateCompleteStatus, [516](#)
  - userInputReq, [516](#)
  - userInputTimeout, [516](#)
  - version, [516](#)
  - versionlength, [516](#)
- omaDmFotaTlvExt, [516](#)
  - description, [518](#)
  - descriptionlength, [518](#)
  - fumoResultCode, [518](#)
  - namelength, [518](#)
  - package\_name, [518](#)
  - packageSize, [518](#)
  - receivedBytes, [518](#)
  - reserved, [518](#)
  - state, [518](#)
  - userInputTimeout, [518](#)
  - version, [518](#)
  - versionlength, [518](#)
- omaDmNotifications
  - sessionInfo, [682](#)
- omaDmNotificationsTlv, [518](#)
  - notification, [518](#)
  - sessionStatus, [518](#)
- Open Mobile Alliance Service (OMA), [41](#)
- operatingMode
  - dms\_OperatingModeTlv, [201](#)
- OperatingModeTlv
  - unpack\_dms\_SetEventReport\_ind\_t, [856](#)
- operation
  - depersonalizationInformation, [195](#)
- OperationMode
  - unpack\_dms\_GetPower\_t, [852](#)
- operatorNameString, [518](#)
  - PLMNName, [519](#)
- OperatorPLMNData, [519](#)
  - lac1, [519](#)
  - lac2, [519](#)
  - mcc, [520](#)
  - mnc, [520](#)
  - PLMNRecID, [520](#)
- operatorPLMNList, [520](#)
  - numInstance, [520](#)
  - PLMNData, [520](#)
- optCode
  - DHCPOption, [198](#)
  - WdsDHCPv4Option, [1057](#)
  - wdsDhcpv4Option, [1058](#)
- optVal
  - WdsDHCPv4Option, [1057](#)
  - wdsDhcpv4Option, [1058](#)
- optValLen
  - DHCPOption, [198](#)
  - WdsDHCPv4Option, [1058](#)
  - wdsDhcpv4Option, [1058](#)
- OriginateUSSD
  - qaGobiApiVoice.h, [1486](#)
- otaMsgTlv
  - NASQmiCbkNasSwiOTAMessageInd, [492](#)
- p3GPP2Pri
  - swiQosFlow, [769](#)
- p3GPP2TimeInfo
  - GetNetworkTimeResp, [255](#)
  - unpack\_nas\_SLQSGetNetworkTime\_t, [894](#)
- p3GPPIImCn
  - swiQosFlow, [769](#)
- p3GPPResResidualBER
  - swiQosFlow, [769](#)
- p3GPPSigInd
  - swiQosFlow, [769](#)
- p3GPPTimeInfo
  - GetNetworkTimeResp, [255](#)
  - unpack\_nas\_SLQSGetNetworkTime\_t, [894](#)
- p3GPPTraHdlPri
  - swiQosFlow, [769](#)
- p3GppNetworkInfoInstances
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- p3GppNetworkInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- p3gppRelease
  - \_slqs3GPPConfigItem, [66](#)
- pAAASPI
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- PACK\_WDS\_IPV4
  - wds.h, [1595](#)
- PACK\_WDS\_IPV6
  - wds.h, [1595](#)
- pAMRStatus
  - voiceGetConfigReq, [1006](#)
- pAPNClass
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pAPNClass3GPP2
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pAPNDisabledFlag
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)

- pAPNEnabled3GPP2
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pAPNName
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
  - qmiWdsRunTimeSettings, [647](#)
  - swiPDPRuntimeSettingsResp, [764](#)
- pAPNnameSize
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- PASSWORD\_LENGTH
  - qaGobiApiVoice.h, [1485](#)
- pAVRXAVCHdroom
  - RXAVCList, [667](#)
- pAVRXAVCSens
  - RXAVCList, [667](#)
- pAccelAcceptReady
  - QmiCbkLocSensorStreamingInd, [636](#)
- pAccelSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [629](#)
- pAccelTempAcceptReady
  - QmiCbkLocSensorStreamingInd, [636](#)
- pAccelTempSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [629](#)
- pAcceleroData
  - LocInjectSensorDataReq, [355](#)
- pAcceleroTempData
  - LocInjectSensorDataReq, [355](#)
- pAcceleroTimeSrc
  - LocInjectSensorDataReq, [355](#)
- pAcqOrder
  - acqOrderPref, [95](#)
  - nas\_acqOrderPref, [385](#)
- pAcqOrderPref
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [548](#)
- pActPilotPNElements
  - NetworkStat1x, [506](#)
- pAddCDMASysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [913](#)
- pAddGSMSysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [913](#)
- pAddHDRSysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [913](#)
- pAddLTESysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [913](#)
- pAddWCDMASysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [913](#)
- pAddrAllocPref
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pAddress
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pAirTimer
  - voiceGetConfigReq, [1006](#)
- pAirTimerCnt
  - voiceGetConfigResp, [1008](#)
- pAirTimerConfig
  - voiceSetConfigReq, [1020](#)
- pAirTimerStatus
  - voiceSetConfigResp, [1022](#)
- pAlertPriority
  - cdmaMsgDecodingParams, [145](#)
- pAlertType
  - voiceCallInfoResp, [985](#)
- pAlertingPattern
  - voiceCallInfoResp, [985](#)
- pAllowLinger
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pAlphaID
  - SMSAsyncRawSend\_s, [732](#)
- pAlphaIDInfo
  - USSResp, [978](#)
  - voiceCallInfoResp, [985](#)
  - voiceCallResponseParams, [988](#)
  - voiceGetCallBarringResp, [995](#)
  - voiceGetCallFWResp, [997](#)
  - voiceGetCallWaitInfo, [998](#)
  - voiceGetCLIPResp, [1000](#)
  - voiceGetCLIRResp, [1001](#)
  - voiceGetCNAPResp, [1002](#)
  - voiceGetCOLPResp, [1003](#)
  - voiceGetCOLRResp, [1005](#)
  - voiceSetCallBarringPwdResp, [1019](#)
  - voiceSetSUPSServiceResp, [1026](#)
  - voiceSUPSInfo, [1029](#)
- pAlphaIdentifier
  - USSDNoWaitIndicationInfo, [977](#)
- pAltitudeAssumed
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [880](#)
- pAltitudeSrcInfo
  - LocInjectPositionReq, [353](#)
- pAltitudeWrtEllipsoid

- LocInjectPositionReq, [353](#)
- PDSPositionData, [584](#)
- QmiCbkLocBestAvailPosInd, [624](#)
- QmiCbkLocPositionReportInd, [635](#)
- unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
- unpack\_loc\_PositionRpt\_Ind\_t, [880](#)
- pAltitudeWrtMeanSeaLevel
  - LocInjectPositionReq, [353](#)
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [880](#)
- pAltitudeWrtSealevel
  - PDSPositionData, [584](#)
- pAmrMode
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pAmrOctetAligned
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pAmrWBMode
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pAmrWBOctetAligned
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pAmrWbEnable
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pApnString
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pApnStringSize
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pApnname
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
- pAppName
  - loc\_LocApplicationInfo, [342](#)
  - LocApplicationInfo, [347](#)
- pAppPriority
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pAppProvider
  - loc\_LocApplicationInfo, [342](#)
  - LocApplicationInfo, [347](#)
- pAppSubType
  - HDRProtSubtypResp, [276](#)
- pAppType
  - LibpackProfile3GPP2, [317](#)
- LibPackprofile\_3GPP2, [328](#)
- Profile3GPP2, [610](#)
- pAppVersion
  - loc\_LocApplicationInfo, [342](#)
  - LocApplicationInfo, [347](#)
- pApplicationInfo
  - LOCStartReq, [357](#)
  - pack\_loc\_Start\_t, [532](#)
- pArrAlertingPattern
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrAlertingType
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrAlphaID
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrCallEndReason
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrCallInfo
  - voiceGetAllCallInfo, [993](#)
- pArrCalledPartyNum
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrConnectPartyNum
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrDiagInfo
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrRedirPartyNum
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrRemotePartyName
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrRemotePartyNum
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrSvcOption
  - voiceGetAllCallInfo, [993](#)
  - voiceSetAllCallStatusCbkInfo, [1017](#)
- pArrUUSInfo
  - voiceGetAllCallInfo, [993](#)
- pAuth
  - pack\_wds\_SLQSSStartDataSession\_t, [578](#)
- pAuthPassword
  - LibpackProfile3GPP2, [317](#)
  - LibPackprofile\_3GPP2, [328](#)
  - Profile3GPP2, [610](#)
- pAuthPassword\_tSize
  - LibPackprofile\_3GPP2, [328](#)
- pAuthPasswordSize
  - LibpackProfile3GPP2, [317](#)
  - Profile3GPP2, [611](#)
- pAuthProtocol
  - LibpackProfile3GPP2, [317](#)

- LibPackprofile\_3GPP2, 328
- Profile3GPP2, 611
- pAuthRetryCount
  - LibpackProfile3GPP2, 318
  - LibPackprofile\_3GPP2, 329
  - Profile3GPP2, 611
- pAuthTimeout
  - LibpackProfile3GPP2, 318
  - LibPackprofile\_3GPP2, 329
  - Profile3GPP2, 611
- pAuthenticateResult
  - UIMAuthenticateResp, 811
- pAuthentication
  - qmiWdsRunTimeSettings, 647
  - ssdatasession\_params, 751
- pAuthenticationPref
  - LibpackProfile3GPP, 312
  - LibPackprofile\_3GPP, 323
  - Profile3GPP, 605
- pAutoAnsStatus
  - voiceSetConfigResp, 1022
- pAutoAnswer
  - voiceGetConfigReq, 1006
  - voiceSetConfigReq, 1021
- pAutoAnswerStat
  - voiceGetConfigResp, 1008
- pAutoSelection
  - UIMGetConfigurationResp, 816
- pAutosdm
  - \_SLQSOMADMS settings, 72
  - \_SLQSOMADMS settings ReqParams, 73
  - \_SLQSOMADMS settings ReqParams3, 74
  - pack\_swima\_SLQSOMADMS setSettings\_t, 559
- pBandPref
  - \_sysSelectPrefInfo, 84
  - \_sysSelectPrefParams, 90
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 548
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 905
- pBdsSVInfo
  - LocDelAssDataReq, 347
  - pack\_loc\_Delete\_Assist\_Data\_t, 527
- pBearerID
  - QosFlowInfo, 651
- pBearerId
  - swiPDPRuntimeSettingsResp, 764
- pBearerTech
  - DataBearerTechExt, 188
- pBurstDTMFLengths
  - voiceBurstDTMFInfo, 982
- pCCResType
  - voiceGetCallBarringResp, 995
  - voiceGetCallFWResp, 997
  - voiceGetCallWaitInfo, 999
  - voiceGetCLIPResp, 1000
  - voiceGetCLIRResp, 1001
  - voiceGetCNAPResp, 1002
  - voiceGetCOLPResp, 1004
  - voiceGetCOLRResp, 1005
  - voiceSetCallBarringPwdResp, 1019
- pCCResultType
  - voiceCallResponseParams, 988
  - voiceSetSUPSServiceResp, 1026
- pCCSUPSType
  - voiceCallResponseParams, 988
  - voiceGetCallBarringResp, 995
  - voiceGetCallFWResp, 997
  - voiceGetCallWaitInfo, 999
  - voiceGetCLIPResp, 1000
  - voiceGetCLIRResp, 1001
  - voiceGetCNAPResp, 1002
  - voiceGetCOLPResp, 1004
  - voiceGetCOLRResp, 1005
  - voiceSetCallBarringPwdResp, 1019
  - voiceSetSUPSServiceResp, 1026
- pCCSuppType
  - USSResp, 978
- pCDMAChannel
  - nasGet3GPP2SubscriptionInfoResp, 470
- pCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 538
  - setSignalStrengthInfo, 704
- pCDMAECIOTresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pCDMAECIOTreshList
  - CDMAECIOTresh, 142
  - nas\_CDMAECIOTresh, 388
- pCDMAFrameErrRate
  - GetErrRateResp, 244
- pCDMAInfo
  - nasCellLocationInfoResp, 468
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pCDMARSSITresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pCDMARSSITreshList
  - CDMARSSITresh, 148
  - nas\_CDMARSSITresh, 390
- pCDMASSInfo
  - nasGetSigInfoResp, 473
- pCDMASigInfo
  - nasSigInfo, 495
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 909
- pCDMASrvStatusInfo
  - nasGetSysInfoResp, 475
  - nasSysInfo, 500
  - unpack\_nas\_SLQSGetSysInfo\_t, 901
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 914
- pCDMASysInfo
  - nasGetSysInfoResp, 475
  - nasSysInfo, 500
  - unpack\_nas\_SLQSGetSysInfo\_t, 901
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 914

- pCLIPResp
  - voiceGetCLIPResp, 1000
- pCLIPstatus
  - voiceSUPSInfo, 1029
- pCLIRCause
  - voiceInfoRec, 1011
- pCLIRResp
  - voiceGetCLIRResp, 1001
- pCLIRType
  - voiceCallRequestParams, 987
- pCLIRstatus
  - voiceSUPSInfo, 1029
- PCMparams, 581
  - iFaceTab, 581
  - iFaceTabLen, 581
- pCNAPResp
  - voiceGetCNAPResp, 1002
- pCNAPstatus
  - voiceSUPSInfo, 1029
- pCOLPResp
  - voiceGetCOLPResp, 1004
- pCOLPstatus
  - voiceSUPSInfo, 1029
- pCOLRResp
  - voiceGetCOLRResp, 1005
- pCOLRstatus
  - voiceSUPSInfo, 1029
- PCSCFAddrPCO
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 967
- PCSCFFQDNAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 967
- PCSCFFQDNAddress, 581
  - fqdnAddr, 581
  - fqdnLen, 582
- PCSCFFQDNAddressList, 582
  - numInstances, 582
  - pcsfFQDNAddress, 582
- PCSCFIPv4ServerAddressList, 582
  - numInstances, 582
  - pcsfIPv4Addr, 583
- pCSCFPortName
  - imsRegMgrConfigInfo, 296
  - SetRegMgrConfigReq, 700
- pCSCFPortNameLen
  - SetRegMgrConfigReq, 700
- pCSGID
  - \_sysSelectPrefParams, 90
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 548
- pCUGIndex
  - voiceSUPSNotification, 1031
- pCUGInfo
  - voiceCallRequestParams, 987
- pCallBarPasswd
  - voiceSUPSInfo, 1029
- pCallBarringPasswd
  - voiceSetSUPSServiceReq, 1025
- pCallEndReason
  - getDUNCallInfoResp, 242
- pCallFWNum
  - voiceSUPSInfo, 1029
- pCallFWTimerVal
  - voiceSUPSInfo, 1029
- pCallForwardingNumber
  - voiceSetSUPSServiceReq, 1025
- pCallFwdInfo
  - voiceSUPSInfo, 1029
- pCallFwdTypeAndPlan
  - voiceSetSUPSServiceReq, 1025
- pCallID
  - burstDTMFInfo, 121
  - voiceCallResponseParams, 988
  - voiceContDTMFInfo, 989
  - voiceFlashInfo, 990
  - voiceGetCallBarringResp, 995
  - voiceGetCallFWResp, 997
  - voiceGetCallWaitInfo, 999
  - voiceGetCLIPResp, 1000
  - voiceGetCLIRResp, 1001
  - voiceGetCNAPResp, 1002
  - voiceGetCOLPResp, 1003
  - voiceGetCOLRResp, 1005
  - voiceManageCallsReq, 1012
  - voiceSetCallBarringPwdResp, 1019
  - voiceSetSUPSServiceResp, 1026
  - voiceSUPSInfo, 1029
- pCallId
  - USSResp, 978
  - voiceAnswerCall, 982
- pCallInfo
  - voiceCallInfoResp, 985
- pCallPartySubAdd
  - voiceCallRequestParams, 987
- pCallType
  - voiceCallRequestParams, 987
- pCallWaitInd
  - voiceInfoRec, 1011
- pCallbackAddr
  - cdmaMsgEncodingParams, 147
- pCallbkAddr
  - cdmaMsgDecodingParams, 145
- pCallbkAddrLength
  - cdmaMsgDecodingParams, 145
- pCalledPartyInfo
  - voiceInfoRec, 1011
- pCallerIDInfo
  - voiceInfoRec, 1011
- pCallerNameInfo
  - voiceInfoRec, 1011
- pCallingPartyInfo
  - voiceInfoRec, 1011
- pCardResult
  - UIMAuthenticateResp, 811
  - UIMGetFileAttributesResp, 817
  - UIMReadTransparentResp, 821
  - unpack\_uim\_ReadTransparent\_t, 951
- pCardStatus

- UIMGetCardStatusResp, [814](#)
- unpack\_uim\_GetCardStatus\_t, [950](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCall-  
Back\_ind\_t, [953](#)
- pCcResultType
  - USSResp, [978](#)
- pCellDb
  - LocDelAssDataReq, [348](#)
  - pack\_loc\_Delete\_Assist\_Data\_t, [527](#)
- pChangeDuration
  - nasInitNetworkReq, [481](#)
  - pack\_nas\_SLQSIInitiateNetworkRegistration\_t, [535](#)
- pChannelRate
  - getDUNCallInfoResp, [242](#)
- pChgDuration
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [548](#)
- pClkInfo
  - LocDelAssDataReq, [348](#)
  - pack\_loc\_Delete\_Assist\_Data\_t, [527](#)
- pCodecSTGain
  - GetAudioPathConfigResp, [232](#)
  - SetAudioPathConfigReq, [685](#)
- pColorCode
  - nasGetHDRColorCodeResp, [471](#)
- pConfidence
  - LocSetCradleMountReq, [356](#)
- pConfigAltitudeAssumed
  - LOCStartReq, [357](#)
  - pack\_loc\_Start\_t, [532](#)
- pConfigurationMask
  - UIMGetConfigurationReq, [815](#)
- pConnectNumInfo
  - voiceCallInfoResp, [985](#)
  - voiceInfoRec, [1011](#)
- pConnectionStatus
  - getDUNCallInfoResp, [242](#)
- pContextId
  - swiPDPRuntimeSettingsResp, [764](#)
- pCrashInfo
  - CrashInfoParams, [168](#)
- pCrashString
  - CrashInfo, [167](#)
- pCreateProfileOut
  - unpack\_wds\_SLQSCreateProfile\_t, [961](#)
- pCurAMRConfig
  - voiceGetConfigResp, [1008](#)
- pCurDataBearerTechnology
  - dataBearers, [185](#)
- pCurPrefVoiceSO
  - voiceGetConfigResp, [1008](#)
- pCurProfile
  - pack\_wds\_SLQSCreateProfile\_t, [571](#)
- pCurVoiceDomainPref
  - voiceGetConfigResp, [1008](#)
- pCurVoicePrivacyPref
  - voiceGetConfigResp, [1008](#)
- pCurrChannelRateInd
  - wdsSetEventReportReq, [1067](#)
- pCurrDataBearerTechInd
  - wdsSetEventReportReq, [1067](#)
- pCurrImgInfo
  - CurrentImgList, [174](#)
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [864](#)
- pCurrNetworkInfo
  - CurrDataSysStat, [172](#)
- pCurrPrefDataSysInd
  - wdsSetEventReportReq, [1067](#)
- pCurrTTYMode
  - voiceGetConfigResp, [1008](#)
- pCurrentChannelRXRate
  - WdsConnectionRateElmnts, [1054](#)
- pCurrentChannelTXRate
  - WdsConnectionRateElmnts, [1054](#)
- pCurrentPersonality
  - HDRPersonalityInd, [275](#)
  - HDRPersonalityResp, [275](#)
- pCurrentPrsnlty
  - HDRProtSubtypResp, [276](#)
- pCurrentmitigationLvl
  - TmdGetMitigationLvlResp, [786](#)
- pCustSettingInfo
  - DMSgetCustomFeatureV2, [204](#)
  - getCustomFeatureV2, [237](#)
- pCustSettingList
  - DMSgetCustomFeatureV2, [204](#)
  - getCustomFeatureV2, [237](#)
- pCwtMute
  - SetM2MAudioProfileReq, [696](#)
  - SetM2MAVMuteReq, [698](#)
- pDDTMInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pDHCPRelayEnabled
  - custFeaturesInfo, [181](#)
  - custFeaturesSetting, [183](#)
- PDOP
  - loc\_precisionDilution, [343](#)
  - precisionDilution\_s, [598](#)
- PDPTYPE
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- pDRCPParams
  - GetHRPDStatsResp, [245](#)
- PDS\_SRV
  - qaGobiApiCbK.h, [1165](#)
- PDSInjectTimeReference
  - qaGobiApiPds.h, [1371](#)
- PDSPosMethodStateReq, [585](#)
- pWifiState, [585](#)
- pXtraDataState, [585](#)
- pXtraTimeState, [585](#)
- PDSPositionData, [583](#)
- pAltitudeWrtEllipsoid, [584](#)
- pAltitudeWrtSealevel, [584](#)
- pHorizontalConfidence, [584](#)
- pHorizontalUncCircular, [584](#)

- pLatitude, [584](#)
- pLongitude, [584](#)
- pPositionSource, [584](#)
- pTimeStamp, [584](#)
- pTimeType, [584](#)
- pVerticalConfidence, [584](#)
- pVerticalUnc, [584](#)
- pDTMFTXGain
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pDataBearer
  - QosEventInfo, [650](#)
  - slqsWdsEventInfo, [730](#)
- pDataBearerTech
  - getDUNCallInfoResp, [242](#)
- pDataBearerTechInd
  - wdsSetEventReportReq, [1067](#)
- pDataCallStatusChangeInd
  - wdsSetEventReportReq, [1067](#)
- pDataMode
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pDataRate
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
  - swiQosFlow, [769](#)
- pDataSrc
  - voiceSUPSInfo, [1029](#)
- pDataStatusDetail
  - NetworkDebugResp, [504](#)
- pDataSystemStatusChangeInd
  - wdsSetEventReportReq, [1067](#)
- pDate
  - \_SLQSOMADMSessionInfo, [70](#)
- pDateLength
  - \_SLQSOMADMSessionInfo, [70](#)
- pDayltSavAdj
  - nasNetworkTime, [483](#)
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind-  
\_t, [908](#)
- pDefaultPDNEnabled
  - \_slqs3GPPConfigItem, [66](#)
- pDeferTime
  - pack\_swioa\_SLQSOMADMSendSelection\_t,  
[558](#)
- pDescription
  - QmiNas3GppNetworkInfo, [642](#)
- pDestAddr
  - cdmaMsgEncodingParams, [147](#)
  - wcdmaMsgEncodingParams, [1037](#)
- pDestSMSContent
  - getDyingGaspCfg, [243](#)
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [524](#)
  - packgetDyingGaspCfg, [580](#)
  - setDyingGaspCfg, [689](#)
- pDestSMSNum
  - getDyingGaspCfg, [243](#)
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [524](#)
  - packgetDyingGaspCfg, [580](#)
  - setDyingGaspCfg, [689](#)
- pDetachAction
  - PSDetachReq, [614](#)
- pDevCrashStatus
  - CrashInfoParams, [168](#)
- pDeviceConfigDetail
  - NetworkDebugResp, [504](#)
- pDiagInfo
  - voiceCallInfoResp, [985](#)
- pDigitBuff
  - burstDTMFInfo, [121](#)
- pDirNum
  - nasGet3GPP2SubscriptionInfoResp, [470](#)
- pDisableIMSI
  - custFeaturesInfo, [181](#)
- pDisplInfo
  - voicInfoRec, [1011](#)
- pDisplayMode
  - cdmaMsgDecodingParams, [145](#)
- pDomainList
  - qmiWdsRunTimeSettings, [647](#)
- pDormancyStatus
  - getDUNCallInfoResp, [242](#)
  - slqsWdsEventInfo, [730](#)
- pDormancyStatusInd
  - wdsSetEventReportReq, [1067](#)
- pDualStandByPrefInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pECIOThresList
  - ECIOThresh, [211](#)
- pECIOThresh
  - sigInfo, [709](#)
- pECMode
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pECTNum
  - voiceSUPSNotification, [1031](#)
- PER
  - NetworkStatEVDO, [508](#)
- pESNString
  - serialNumbersInfo, [678](#)
- pEVDOPageMonPerChangeInd
  - wdsSetEventReportReq, [1067](#)
- pEarMute
  - SetM2MAudioProfileReq, [696](#)
- pEmerMode
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [548](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [905](#)
- pEmergencyCategory
  - voiceCallRequestParams, [987](#)
- pEnableNotification
  - WdsClientLeaseChange, [1052](#)

- pEnabled
  - pack\_wds\_SetMobileIPProfile\_t, 570
- pEncodingAlphabet
  - cdmaMsgEncodingParams, 147
- pEncryptData
  - pack\_uim\_ReadTransparent\_t, 561
  - UIMReadTransparentReq, 820
- pEncryptedData
  - UIMReadTransparentResp, 821
  - unpack\_uim\_ReadTransparent\_t, 951
- pEncryptedPIN1
  - pack\_uim\_VerifyPin\_t, 566
  - UIMPinResp, 818
  - UIMVerifyPinReq, 832
  - unpack\_uim\_ChangePin\_t, 949
  - unpack\_uim\_SetPinProtection\_t, 951
  - unpack\_uim\_UnblockPin\_t, 954
  - unpack\_uim\_VerifyPin\_t, 955
- pError
  - USSDNoWaitIndicationInfo, 977
- pErrorCodeStr
  - imsaRatStatusInfo, 289
- pErrorMask
  - CATEventDataType, 138
- pErrorRateInd
  - nasIndicationRegisterReq, 480
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 542
- pEspSpi
  - swiQosFilter, 766
- pEtwsMessageInfo
  - SMSEventInfo\_s, 737
- pEtwsPlmnInfo
  - SMSEventInfo\_s, 737
- pExtDispInfo
  - voiceInfoRec, 1011
- pExtDispRecInfo
  - voiceInfoRec, 1011
- pExtErrCode
  - \_GetProfileSettingOut, 57
  - UnPackGetProfileSettingOut, 973
- pExtErrorCode
  - CreateProfileOut, 169
  - ModifyProfileOut, 383
  - unpack\_wds\_SLQSMModifyProfile\_t, 967
- pFOTAUpdate
  - \_SLQSOMADMSSettings, 72
- pFOTAdownload
  - \_SLQSOMADMSSettings, 72
- pFailCause
  - voiceGetCallBarringResp, 995
  - voiceGetCallFWResp, 997
  - voiceGetCallWaitInfo, 999
  - voiceGetCLIPResp, 1000
  - voiceGetCLIRResp, 1001
  - voiceGetCNAPResp, 1002
  - voiceGetCOLPResp, 1004
  - voiceGetCOLRResp, 1005
  - voiceManageCallsResp, 1013
  - voiceSetCallBarringPwdResp, 1019
  - voiceSetSUPSServiceResp, 1026
  - voiceSUPSInfo, 1029
- pFailErrorCode
  - imsaPdpStatusInfo, 288
- pFailureCause
  - USSDNoWaitIndicationInfo, 977
- pFailureReason
  - unpack\_wds\_SLQSStartDataSession\_t, 971
- pFile
  - ERIFileparams, 213
- pFileAttributes
  - UIMGetFileAttributesResp, 817
- pFileSize
  - ERIFileparams, 213
- pFixId
  - QmiCbkLocPositionReportInd, 635
  - unpack\_loc\_PositionRpt\_Ind\_t, 880
- pFlag
  - RXPCMIIRFiltr, 670
  - TXPCMIIRFiltr, 795
- pFlashPayLd
  - voiceFlashInfo, 990
- pFlashType
  - voiceFlashInfo, 990
- pFollowOnDC
  - slqssendasyncsmsparams\_s, 721
- pForbidden
  - QmiNas3GppNetworkInfo, 642
- pForceOnDC
  - slqssendasyncsmsparams\_s, 721
- pFwAutoCheck
  - \_SLQSOMADMSSettings, 72
  - \_SLQSOMADMSSettingsReqParams3, 74
  - pack\_swima\_SLQSOMADMSSetSettings\_t, 559
- pGCDumpString
  - CrashInfo, 167
- pGERANInfo
  - nasCellLocationInfoResp, 469
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pGPRSGrantedQoS
  - qmiWdsRunTimeSettings, 647
- pGPRSMInimumQoS
  - LibpackProfile3GPP, 312
  - LibPackprofile\_3GPP, 323
  - Profile3GPP, 605
- pGPRSRequestedQoS
  - LibpackProfile3GPP, 312
  - LibPackprofile\_3GPP, 323
  - Profile3GPP, 605
- pGPSEnable
  - custFeaturesSetting, 183
- pGPSLPM
  - custFeaturesInfo, 181
  - custFeaturesSetting, 183
- pGPSSel
  - custFeaturesInfo, 181
  - custFeaturesSetting, 183

- pGSMBER
  - GetErrRateResp, [244](#)
- pGSMCallBarringSysInfo
  - nasGetSysInfoResp, [475](#)
  - nasSysInfo, [500](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pGSMCipherDomainSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pGSMRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pGSMRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pGSMRSSIThreshList
  - GSMRSSIThresh, [268](#)
  - nas\_GSMRSSIThresh, [407](#)
- pGSMSSInfo
  - nasGetSigInfoResp, [473](#)
- pGSMSSigInfo
  - nasSigInfo, [495](#)
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [909](#)
- pGSMSSrvStatusInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pGSMSSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pGWAcqOrderPref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [548](#)
  - unpack\_nas\_SLQSSetSysSelectionPref\_t, [905](#)
- pGWAddressV4
  - qmiWdsRunTimeSettings, [647](#)
- pGenerator
  - GetM2MAudioProfileReq, [250](#)
  - SetM2MAudioProfileReq, [696](#)
- pGetCallFWExtInfo
  - voiceGetCallFWResp, [997](#)
- pGetCallFWInfo
  - voiceGetCallFWResp, [997](#)
- pGetCustomInput
  - DMSgetCustomFeatureV2, [204](#)
  - getCustomFeatureV2, [237](#)
- pGetDyingGaspCfg
  - unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, [863](#)
- pGetDyingGaspStatistics
  - unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, [863](#)
- pGnssData
  - LocDelAssDataReq, [348](#)
  - pack\_loc\_Delete\_Assist\_Data\_t, [527](#)
- pGpsTime
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [880](#)
- pGyroAcceptReady
  - QmiCbkLocSensorStreamingInd, [636](#)
- pGyroData
  - LocInjectSensorDataReq, [355](#)
- pGyroSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [629](#)
- pGyroTempAcceptReady
  - QmiCbkLocSensorStreamingInd, [636](#)
- pGyroTempData
  - LocInjectSensorDataReq, [355](#)
- pGyroTempSamplesAccepted
  - QmiCbkLocInjectSensorDataInd, [629](#)
- pGyroTimeSrc
  - LocInjectSensorDataReq, [355](#)
- pHASPI
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pHDRECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHDRECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHDRECIOThreshList
  - HDRECIOThresh, [274](#)
  - nas\_HDRECIOThresh, [411](#)
- pHDRIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHDRIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHDRIOThreshList
  - HDRIOThresh, [274](#)
  - nas\_HDRIOThresh, [412](#)
- pHDRNewUATIAssInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pHDRPackErrRate
  - GetErrRateResp, [244](#)
- pHRRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHRRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
- pHRRSSIThreshList
  - HRRSSIThresh, [277](#)
  - nas\_HDRSSIThresh, [412](#)
- pHRSINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)

- setSignalStrengthInfo, [704](#)
- pHRSINRThresList
  - HRSINRThresh, [277](#)
- pHRSINRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [704](#)
  - sigInfo, [709](#)
- pHRSINRThreshList
  - HRSINRThreshold, [278](#)
  - nas\_HRSINRThreshold, [413](#)
- pHRSInfo
  - nasGetSigInfoResp, [473](#)
- pHRSsessionCloseInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pHRSigInfo
  - nasSigInfo, [496](#)
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [909](#)
- pHRSrvStatusInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pHRSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pHaltSubscription
  - UIMGetConfigurationResp, [816](#)
- pHeading
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [880](#)
- pHeadingUnc
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHomeSIDNID
  - nasGet3GPP2SubscriptionInfoResp, [470](#)
- pHorCirConf
  - QmiCbkLocBestAvailPosInd, [624](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
- pHorConfidence
  - LocInjectPositionReq, [353](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorEllpConf
  - QmiCbkLocBestAvailPosInd, [624](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
- pHorReliability
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorUncCircular
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorUncEllipseOrientAzimuth
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorUncEllipseSemiMajor
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorUncEllipseSemiMinor
  - QmiCbkLocBestAvailPosInd, [624](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [873](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pHorizontalAccuracyLvl
  - LOCStartReq, [357](#)
  - pack\_loc\_Start\_t, [532](#)
- pHorizontalConfidence
  - PDSPositionData, [584](#)
- pHorizontalUncCircular
  - PDSPositionData, [584](#)
- pHotSwapStatus
  - UIMGetCardStatusResp, [814](#)
  - unpack\_uim\_GetCardStatus\_t, [950](#)
- pHwConfig
  - unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, [970](#)
  - WdsDHCPv4Config, [1056](#)
- PI
  - calledPartyInfo, [125](#)
  - callerIDInfo, [126](#)
  - callFWExtInfo, [129](#)
  - callingPartyInfo, [133](#)
  - redirNumInfo, [656](#)
- PIFACEId
  - SetM2MAudioAVCFGRReq, [694](#)
- pIMCNflag
  - qmiWdsRunTimeSettings, [647](#)
- pIMEIString
  - serialNumbersInfo, [678](#)
- pIMSDomain
  - GetIMSUserConfigParams, [247](#)
  - imsUserConfigInfo, [298](#)
  - SetIMSUserConfigReq, [690](#)
- pIMSDomainLen
  - GetIMSUserConfigParams, [247](#)
  - SetIMSUserConfigReq, [691](#)
- pIMSTestMode
  - GetRegMgrConfigParams, [256](#)
  - imsRegMgrConfigInfo, [296](#)
  - SetRegMgrConfigReq, [700](#)

- plOThresList
  - IOThresh, [303](#)
- plOThresh
  - sigInfo, [709](#)
- piPAddressV4
  - QmiWdsIpAddressInfo, [644](#)
  - qmiWdsRunTimeSettings, [647](#)
- piPAddressV6
  - QmiWdsIpAddressInfo, [644](#)
- piPFamSupport
  - custFeaturesInfo, [181](#)
- piPFamily
  - GetInstIDResp, [249](#)
- piPFamilyPreference
  - qmiWdsRunTimeSettings, [647](#)
- piPV6AddrInfo
  - qmiWdsRunTimeSettings, [647](#)
- piPV6GWAddrInfo
  - qmiWdsRunTimeSettings, [647](#)
- piPv4Addr
  - WdsDHCPv4ClientLeaseInd, [1055](#)
- piPv4AddrPref
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- piPv4Address
  - swiPDPRuntimeSettingsResp, [764](#)
- piPv4DstAddr
  - swiQosFilter, [766](#)
- piPv4GWAddress
  - swiPDPRuntimeSettingsResp, [764](#)
- piPv4SrcAddr
  - swiQosFilter, [766](#)
- piPv6AddPref
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- piPv6Address
  - swiPDPRuntimeSettingsResp, [764](#)
- piPv6DstAddr
  - swiQosFilter, [766](#)
- piPv6GWAddress
  - swiPDPRuntimeSettingsResp, [764](#)
- piPv6Label
  - swiQosFilter, [766](#)
- piPv6SrcAddr
  - swiQosFilter, [766](#)
- piPv6TrafCls
  - swiQosFilter, [766](#)
- piPv6prefixlen
  - QmiWdsIpAddressInfo, [644](#)
- pld
  - swiQosFilter, [766](#)
- plds
  - swiQosIds, [770](#)
- plgnoreHotSwapSwitch
  - UIMPowerUpReq, [819](#)
- plmCnFlag
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- plmgeList
  - pack\_fms\_SetImagesPreference\_t, [526](#)
  - unpack\_fms\_GetImagesPreference\_t, [867](#)
- plmeiSvnString
  - serialNumbersInfo, [678](#)
- plmgType
  - FirmwareUpdatStat, [221](#)
- plmsRegErrCode
  - IMSARegistrationStatus, [290](#)
- plmsRegStatus
  - IMSARegistrationStatus, [290](#)
  - imsaRegStatusInfo, [290](#)
- plnUse
  - QmiNas3GppNetworkInfo, [642](#)
- plndFieldsList
  - IMSASupportedFieldsResp, [293](#)
- plndicationToken
  - pack\_uim\_ChangePin\_t, [560](#)
  - pack\_uim\_ReadTransparent\_t, [561](#)
  - pack\_uim\_SetPinProtection\_t, [562](#)
  - pack\_uim\_UnblockPin\_t, [565](#)
  - pack\_uim\_VerifyPin\_t, [566](#)
  - UIMAuthenticateReq, [811](#)
  - UIMAuthenticateResp, [811](#)
  - UIMChangePinReq, [812](#)
  - UIMGetFileAttributesReq, [816](#)
  - UIMGetFileAttributesResp, [817](#)
  - UIMPinResp, [818](#)
  - UIMReadTransparentReq, [820](#)
  - UIMReadTransparentResp, [821](#)
  - UIMSetPinProtectionReq, [827](#)
  - UIMUnblockPinReq, [831](#)
  - UIMVerifyPinReq, [832](#)
  - unpack\_uim\_ChangePin\_t, [949](#)
  - unpack\_uim\_ReadTransparent\_t, [951](#)
  - unpack\_uim\_SetPinProtection\_t, [951](#)
  - unpack\_uim\_UnblockPin\_t, [954](#)
  - unpack\_uim\_VerifyPin\_t, [955](#)
- plnstanceID
  - GetInstIDResp, [249](#)
- plnstanceSize
  - QmiNasPerformNetworkScanResp, [643](#)
- plnstances
  - QmiNasPerformNetworkScanResp, [643](#)
- plnstancesSize
  - QmiNasGetRFBandInfoResp, [642](#)
- plntermediateReportState
  - LOCStartReq, [358](#)
  - pack\_loc\_Start\_t, [532](#)
- plpcpAckTimeout
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- plpcpCreqRetryCount
  - LibpackProfile3GPP2, [318](#)

- LibPackprofile\_3GPP2, 329
- Profile3GPP2, 611
- plsPcscfAddressNedded
  - LibpackProfile3GPP2, 318
  - LibPackprofile\_3GPP2, 329
  - Profile3GPP2, 611
- plsUDHPresent
  - wcdmaLongMsgDecodingParams, 1034
- plsVoiceEnabled
  - custFeaturesInfo, 181
  - custFeaturesSetting, 183
- pJitter
  - swiQosFlow, 769
- pKeyReferenceID
  - pack\_uim\_ChangePin\_t, 560
  - pack\_uim\_SetPinProtection\_t, 563
  - pack\_uim\_UnblockPin\_t, 565
  - pack\_uim\_VerifyPin\_t, 566
  - UIMChangePinReq, 812
  - UIMSetPinProtectionReq, 827
  - UIMUnblockPinReq, 831
  - UIMVerifyPinReq, 832
- PLMN\_LENGTH
  - qaGobiApiNas.h, 1322
- PLMNData
  - operatorPLMNList, 520
- PLMNName
  - operatorNameString, 519
- PLMNNetName
  - PLMNNetworkName, 595
- PLMNNetworkName, 595
  - numInstance, 595
  - PLMNNetName, 595
- PLMNNetworkNameData, 595
  - codingScheme, 596
  - countryInitials, 596
  - longName, 596
  - longNameLen, 596
  - longNameSpareBits, 597
  - shortName, 597
  - shortNameLen, 597
  - shortNameSpareBits, 597
- PLMNRecID
  - OperatorPLMNData, 520
- pLTEAttachProfile
  - \_slqs3GPPConfigItem, 66
- pLTEAttachProfileList
  - \_slqs3GPPConfigItem, 66
- pLTEBandPref
  - \_sysSelectPrefInfo, 84
  - \_sysSelectPrefParams, 90
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 548
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 905
- pLTECphyCa
  - nasIndicationRegisterReq, 480
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 542
- pLTEInfo
  - swiModemStatusResp, 760
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, 909
- pLTEInfoInterfreq
  - nasCellLocationInfoResp, 469
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pLTEInfoIntrafreq
  - nasCellLocationInfoResp, 469
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pLTEInfoNeighboringGSM
  - nasCellLocationInfoResp, 469
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pLTEInfoNeighboringWCDMA
  - nasCellLocationInfoResp, 469
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 906
- pLTERSRPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pLTERSRPThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pLTERSRPThreshList
  - LTERSRPThresh, 369
  - nas\_LTERSRPThresh, 425
- pLTERSRQDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pLTERSRQThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 704
- pLTERSRQThreshList
  - LTERSRQThresh, 370
  - nas\_LTERSRQThresh, 425
- pLTERSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 705
- pLTERSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 705
- pLTERSSIThreshList
  - LTERSSIThresh, 370
  - nas\_LTERSSIThresh, 426
- pLTERSNRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 705
- pLTERSNRThresList
  - LTERSNRThresh, 373
- pLTERSNRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 539
  - setSignalStrengthInfo, 705
  - sigInfo, 709
- pLTERSNRThreshList
  - LTERSNRThreshold, 374
  - nas\_LTERSNRThreshold, 427
- pLTERSSInfo
  - nasGetSigInfoResp, 473
- pLTERSigInfo
  - nasSigInfo, 496
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 909
- pLTERSigRptCfg

- sigInfo, [709](#)
- pLTESigRptConfig
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pLTESrvStatusInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pLTESysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pLTEVoiceSupportSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pLanguage
  - cdmaMsgDecodingParams, [145](#)
- pLastBearerTech
  - DataBearerTechExt, [188](#)
- pLastCallDataBearerTech
  - getDUNCallInfoResp, [242](#)
- pLastCallDataBearerTechnology
  - dataBearers, [185](#)
- pLastCallRXOKBytesCnt
  - getDUNCallInfoResp, [242](#)
- pLastCallTXOKBytesCnt
  - getDUNCallInfoResp, [242](#)
- pLatency
  - swiQosFlow, [769](#)
- pLatitude
  - LocInjectPositionReq, [354](#)
  - PDSPositionData, [584](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pLcpAckTimeout
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pLcpCreqRetryCount
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pLeapSeconds
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pLeaseState
  - WdsDHCPv4ClientLeaseInd, [1055](#)
- pLineCtrlInfo
  - voicelInfoRec, [1011](#)
- pLinktimer
  - pack\_sms\_SendSMS\_t, [551](#)
  - slqssendasyncsmsparams\_s, [721](#)
- slqssendsmsparams\_s, [722](#)
- pLogString
  - FirmwareUpdatStat, [221](#)
- pLogStringLength
  - FirmwareUpdatStat, [222](#)
- pLongitude
  - LocInjectPositionReq, [354](#)
  - PDSPositionData, [584](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pLoopbackMode
  - WDSsetLoopbackData, [1067](#)
- pLoopbackMultiplier
  - WDSsetLoopbackData, [1067](#)
- pLteBandCapability
  - BandCapabilityResp, [119](#)
- pLteEARFCN
  - nasSwiGetChannelLockResp, [496](#)
  - nasSwiSetChannelLockReq, [498](#)
- pLteNasRelInfo
  - SwiOTAMsg\_s, [761](#)
- pLtePCI
  - nasSwiGetChannelLockResp, [496](#)
  - nasSwiSetChannelLockReq, [498](#)
- pLteQci
  - swiQosFlow, [769](#)
- pMCC
  - QmiNas3GppNetworkInfo, [642](#)
- pMEIDString
  - serialNumbersInfo, [678](#)
- pMICGainSelect
  - GetAudioPathConfigResp, [233](#)
- pMIPStatusInd
  - wdsSetEventReportReq, [1067](#)
- pMNAAB
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pMNC
  - QmiNas3GppNetworkInfo, [642](#)
- pMNCIncPCSDigStat
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
- pMNHA
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pMNRInfo
  - nasInitNetworkReg, [481](#)
  - pack\_nas\_SLQSSetNetworkRegistration\_t, [535](#)
- pMTMessageInfo
  - SMSEventInfo\_s, [738](#)
- pMagneticDeviation
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pManString
  - \_SLQSSwiGetHostDevInfoParams, [75](#)
  - \_SLQSSwiSetHostDevInfoParams, [78](#)

- pManagedRoamingInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pMaxAllowedPktSz
  - swiQosFlow, [769](#)
- pMaxChannelRXRate
  - WdsConnectionRateElmnts, [1054](#)
- pMaxChannelTXRate
  - WdsConnectionRateElmnts, [1054](#)
- pMdmCallDurationActive
  - getDUNCallInfoResp, [242](#)
- pMeidString
  - \_SLQSSwiGetSerialNoExtParams, [77](#)
- pMessage
  - cdmaMsgDecodingParams, [146](#)
  - cdmaMsgEncodingParams, [147](#)
  - pack\_sms\_SendSMS\_t, [551](#)
  - slqssendasyncsmsparams\_s, [721](#)
  - slqssendsmsparams\_s, [722](#)
  - wcdmaLongMsgDecodingParams, [1034](#)
  - wcdmaMsgDecodingParams, [1036](#)
- pMessageID
  - cdmaMsgDecodingParams, [146](#)
- pMessageIndex
  - pack\_sms\_SLQSDDeleteSMS\_t, [552](#)
- pMessageMode
  - pack\_sms\_SLQSDDeleteSMS\_t, [553](#)
  - pack\_sms\_SLQSGetSMS\_t, [553](#)
  - pack\_sms\_SLQSGetSMSList\_t, [554](#)
  - pack\_sms\_SLQSMModifySMSStatus\_t, [555](#)
  - smsMaxStorageSizeReq, [738](#)
- pMessageModelInfo
  - SMSEventInfo\_s, [737](#)
- pMessageSize
  - cdmaMsgEncodingParams, [147](#)
- pMessageTag
  - pack\_sms\_SLQSDDeleteSMS\_t, [553](#)
- pMicMute
  - SetM2MAudioProfileReq, [696](#)
- pMinBasedIMSI
  - nasGet3GPP2SubscriptionInfoResp, [470](#)
- pMinIntervalTime
  - LOCStartReq, [358](#)
  - pack\_loc\_Start\_t, [532](#)
- pMinPolicedPktSz
  - swiQosFlow, [770](#)
- pMinSessionExpiryTimer
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pMitigationDevList
  - TmdGetMitigationDevListResp, [785](#)
- pMitigationDevListLen
  - TmdGetMitigationDevListResp, [785](#)
- pMncPcsDigitStatus
  - nasInitNetworkReg, [481](#)
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, [535](#)
- pMncPcsStatus
  - nasPLMNNameReq, [489](#)
  - pack\_nas\_SLQSGetPLMNName\_t, [534](#)
- pModePref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [905](#)
- pModelString
  - \_SLQSSwiGetHostDevInfoParams, [75](#)
  - \_SLQSSwiSetHostDevInfoParams, [78](#)
- pMtu
  - qmiWdsRunTimeSettings, [647](#)
- pNAI
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pNAMNameInfo
  - nasGet3GPP2SubscriptionInfoResp, [470](#)
- pNITZInformation
  - nasOperatorNameResp, [484](#)
- pNSEnable
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pNSSAudioCtrl
  - voiceInfoRec, [1011](#)
- pNSSRelease
  - voiceInfoRec, [1011](#)
- pNamID
  - voiceGetConfigReq, [1006](#)
- pName
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
- pNameString
  - \_SLQSSwiGetOSInfoParams, [76](#)
  - \_SLQSSwiSetOSInfoParams, [79](#)
- pNegoDnsSrvrPref
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pNeighborSetPilotPN
  - NetworkStat1x, [506](#)
- pNetSelPref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [905](#)
- pNetworkInfo
  - \_slqsNetworkScanInfo, [68](#)
- pNetworkInfoInstances
  - \_slqsNetworkScanInfo, [68](#)
- pNetworkInfoLen
  - CurrDataSysStat, [172](#)
- pNetworkStat1x
  - NetworkDebugResp, [504](#)
- pNetworkStatEVDO
  - NetworkDebugResp, [504](#)
- pNetworkTimeInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pNewImsRegStatus
  - IMSARegistrationStatus, [290](#)

- pNewPwdData
  - voiceSUPSInfo, [1029](#)
- pNumSupUSBComps
  - USBCompParams, [976](#)
- pNumberOfPhySlot
  - UIMGetSlotsStatusResp, [817](#)
  - unpack\_uim\_SLQSUIGetSlotsStatus\_t, [953](#)
- pNxtHdrProto
  - swiQosFilter, [766](#)
- pOMADMEnabled
  - \_SLQSOMADMSettings, [72](#)
- PORTNAM\_LEN
  - qaGobiApiDcs.h, [1238](#)
- pOTASPStatus
  - voiceCallInfoResp, [985](#)
  - voiceGetAllCallInfo, [993](#)
- pObjectVer
  - NetworkDebugResp, [504](#)
- pOffLength
  - voiceDTMFEventInfo, [990](#)
- pOnLength
  - voiceDTMFEventInfo, [990](#)
- pOpaqueIdentifier
  - LocInjectSensorDataReq, [355](#)
  - QmiCbkLocInjectSensorDataInd, [629](#)
- pOperatorNameString
  - nasOperatorNameResp, [484](#)
- pOperatorPLMNList
  - nasOperatorNameResp, [484](#)
- pOptList
  - WdsDHCPv4ClientLeaseInd, [1055](#)
  - WdsDHCPv4OptionList, [1059](#)
  - wdsDhcpv4OptionList, [1058](#)
- pOptVal
  - DHCPOption, [198](#)
- pOptions
  - DHCPOptionList, [199](#)
- pPCMPParams
  - SetM2MAudioAVCFGReq, [694](#)
- pPCSCFAddrPCO
  - qmiWdsRunTimeSettings, [647](#)
- pPCSCFFQDNAddrList
  - qmiWdsRunTimeSettings, [647](#)
- pPCSCFPort
  - GetRegMgrConfigParams, [256](#)
- pPCSDigitInfo
  - \_slqsNetworkScanInfo, [68](#)
- pPCSDigitInstances
  - \_slqsNetworkScanInfo, [68](#)
- pPCSIInstance
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- pPCSIInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- pPDNInactivTimeout
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPDNInactivTimeout3GPP2
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPDType
  - qmiWdsRunTimeSettings, [647](#)
- pPDPtype
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [606](#)
- pPDUMessage
  - wcdmaMsgEncodingParams, [1037](#)
- pPLMNNetworkName
  - nasOperatorNameResp, [484](#)
- pPRLPref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [905](#)
- pPRLPreference
  - dmsCurrentPRLInfo, [202](#)
- pPRLVersion
  - dmsCurrentPRLInfo, [202](#)
- pPacketsCountRX
  - QosEventInfo, [650](#)
  - slqsWdsEventInfo, [730](#)
- pPacketsCountTX
  - QosEventInfo, [650](#)
  - slqsWdsEventInfo, [730](#)
- pPartNum
  - wcdmaLongMsgDecodingParams, [1035](#)
- pPass
  - pack\_wds\_SLQSStartDataSession\_t, [578](#)
- pPassword
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
  - Profile3GPP, [605](#)
  - ssdatasession\_params, [751](#)
- pPasswordSize
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPcsfAddrUsingDhcp
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPcsfAddrUsingPCO
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPdnType
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPdpAccessConFlag
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)

- pPdpContext
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPdpDataCompType
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [605](#)
- pPdpHdrCompType
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [606](#)
- pPdpStatusConfig
  - IMSAIndRegisterInfo, [287](#)
- pPersonalityListLength
  - HDRPersonalityInd, [275](#)
  - HDRPersonalityResp, [275](#)
  - HDRProtSubtypResp, [276](#)
- pPersonalizationStatus
  - UIMGetConfigurationResp, [816](#)
- pPhoneCtxtURI
  - GetIMSSMSConfigParams, [246](#)
  - imsSMSConfigInfo, [298](#)
  - SetIMSSMSConfigReq, [690](#)
- pPhoneCtxtURLen
  - GetIMSSMSConfigParams, [246](#)
  - SetIMSSMSConfigReq, [690](#)
- pPhyCaAggPcellInfo
  - nasGetLTECphyCaResp, [472](#)
- pPhyCaAggScellDIBw
  - nasGetLTECphyCaResp, [472](#)
- pPhyCaAggScellIndType
  - nasGetLTECphyCaResp, [472](#)
- pPhyCaAggScellIndex
  - nasGetLTECphyCaResp, [472](#)
- pPhyCaAggScellInfo
  - nasGetLTECphyCaResp, [472](#)
- pPilotSetData
  - GetHRPDStatsResp, [245](#)
- pPilotSetInfo
  - PilotSetData, [593](#)
- pPkgDescLength
  - \_SLQSOMADMSessionInfo, [70](#)
- pPkgDescription
  - \_SLQSOMADMSessionInfo, [70](#)
- pPkgName
  - \_SLQSOMADMSessionInfo, [70](#)
- pPkgNameLength
  - \_SLQSOMADMSessionInfo, [70](#)
- pPktErrRate
  - swiQosFlow, [770](#)
- pPlasmaIDString
  - \_SLQSSwiGetHostDevInfoParams, [75](#)
  - \_SLQSSwiSetHostDevInfoParams, [78](#)
- pPositionSource
  - PDSPositionData, [584](#)
- pPositionSrc
  - LocInjectPositionReq, [354](#)
- pPppSessCloseTimer1x
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPppSessCloseTimerDO
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPrDNSIPv4Address
  - swiPDPRuntimeSettingsResp, [764](#)
- pPrDNSIPv6Address
  - swiPDPRuntimeSettingsResp, [764](#)
- pPrPCSCFIPv4Address
  - swiPDPRuntimeSettingsResp, [764](#)
- pPrPCSCFIPv6Address
  - swiPDPRuntimeSettingsResp, [764](#)
- pPrecedence
  - swiQosFilter, [766](#)
- pPrecisionDilution
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pPrefNetwork
  - CurrDataSysStat, [172](#)
- pPrefVoiceDomain
  - voiceSetConfigReq, [1021](#)
- pPrefVoicePrivacy
  - voiceGetConfigReq, [1006](#)
- pPrefVoiceSO
  - voiceGetConfigReq, [1006](#)
  - voiceSetConfigReq, [1021](#)
- pPrefVoiceSOStatus
  - voiceSetConfigResp, [1022](#)
- pPreferred
  - QmiNas3GppNetworkInfo, [642](#)
- pPriCSCFPort
  - imsRegMgrConfigInfo, [296](#)
  - SetRegMgrConfigReq, [700](#)
- pPriCSCFPortName
  - GetRegMgrConfigParams, [256](#)
- pPriCSCFPortNameLen
  - GetRegMgrConfigParams, [256](#)
- pPriDNSIPv4AddPref
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [606](#)
- pPriDNSIPv6addpref
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [606](#)
- pPriV6DnsAddress
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPrimaryDNSV4
  - qmiWdsRunTimeSettings, [648](#)
- pPrimaryDNSV6

- qmiWdsRunTimeSettings, [648](#)
- pPrimaryHA
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pPrimaryID
  - LibpackProfile3GPP, [312](#)
  - LibPackprofile\_3GPP, [323](#)
  - Profile3GPP, [606](#)
- pPrimaryV4DnsAddress
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pPriority
  - cdmaMsgDecodingParams, [146](#)
  - cdmaMsgEncodingParams, [147](#)
- pPrivacy
  - cdmaMsgDecodingParams, [146](#)
- pProfSz
  - NWProfile, [511](#)
- pProfValues
  - NWProfile, [511](#)
- pProfileID
  - CreateProfileIn, [169](#)
  - ModifyProfileIn, [383](#)
  - qmiWdsRunTimeSettings, [648](#)
  - unpack\_wds\_SLQSCreateProfile\_t, [961](#)
- pProfileId
  - pack\_wds\_SLQSCreateProfile\_t, [571](#)
  - pack\_wds\_SLQSModifyProfile\_t, [575](#)
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [577](#)
  - WdsDHCPv4ClientLeaseInd, [1055](#)
  - WdsDHCPv4Config, [1056](#)
- pProfileId3GPP
  - ssdatasession\_params, [751](#)
- pProfileId3GPP2
  - ssdatasession\_params, [751](#)
  - swiQosFlow, [770](#)
- pProfileIndex
  - CreateProfileOut, [169](#)
- pProfileList
  - \_slqs3GPPConfigItem, [66](#)
- pProfileName
  - qmiWdsRunTimeSettings, [648](#)
- pProfileSettings
  - unpack\_wds\_SLQSGetProfileSettings\_t, [965](#)
- pProfileType
  - CreateProfileIn, [169](#)
  - CreateProfileOut, [169](#)
  - ModifyProfileIn, [383](#)
  - pack\_wds\_SLQSCreateProfile\_t, [571](#)
  - pack\_wds\_SLQSModifyProfile\_t, [575](#)
- pProfilename
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pProfilenameSize
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pProtoSubTypElmnt
  - HDRProtSubtypResp, [276](#)
- pProtocolSubtypeElement
  - HDRPersonalityInd, [275](#)
  - HDRPersonalityResp, [275](#)
- pQFlowState
  - QosFlowInfo, [651](#)
- pQmiInterfaceInfo
  - \_packetSrvStatus, [62](#)
  - slqsSessionStateInfo, [723](#)
  - slqsWdsEventInfo, [730](#)
- pQosClassID
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pRAT
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
- pRATInstance
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- pRATInfo
  - \_slqsNetworkScanInfo, [68](#)
- pRATInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, [890](#)
- pRATInstances
  - \_slqsNetworkScanInfo, [68](#)
- pRATStatus
  - imsaRatStatusInfo, [289](#)
- pRATType
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- PREFERRED\_INDEX
  - qaNasPerformNetworkScan.h, [1546](#)
- pRFBandInfoElements
  - QmiNasGetRFBandInfoResp, [642](#)
- PRI\_UPDATE\_FAIL
  - qaGobiApiFms.h, [1287](#)
- PRIString
  - unpack\_dms\_GetFirmwareRevision\_t, [849](#)
  - unpack\_dms\_GetFirmwareRevisions\_t, [850](#)
- PRLInd
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- PRLPTlv
  - NASQmiCbkNasSystemSelPrefInd, [493](#)
- PRLPref
  - NASPRLPreferenceTlv, [492](#)
- pRMAutoConnect
  - custFeaturesInfo, [181](#)
- pRSRPThresList
  - RSRPThresh, [664](#)
- pRSRPThresh
  - sigInfo, [709](#)
- pRSRQThresList
  - RSRQThresh, [665](#)
- pRSRQThresh
  - sigInfo, [709](#)

- pRSSIThresList
  - RSSIThresh, [666](#)
- pRSSIThresh
  - sigInfo, [709](#)
- pRTPRTCPIInactTimer
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pRXAGCList
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pRXAIG
  - RXAGCList, [667](#)
- pRXAVCAGCSwitch
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pRXAVCList
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pRXChain0Info
  - nasGetTxRxInfoResp, [477](#)
- pRXChain1Info
  - nasGetTxRxInfoResp, [477](#)
- pRXComprSlope
  - RXAGCList, [667](#)
- pRXComprThres
  - RXAGCList, [667](#)
- pRXDroppedCount
  - WdsPktStatisticsElmnts, [1062](#)
- pRXExpSlope
  - RXAGCList, [667](#)
- pRXExpThres
  - RXAGCList, [667](#)
- pRXOKBytesCount
  - getDUNCallInfoResp, [242](#)
- pRXOKBytesLastCall
  - WdsPktStatisticsElmnts, [1062](#)
- pRXOkBytesCount
  - WdsPktStatisticsElmnts, [1062](#)
- pRXPCMIIRFtr
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pRXPacketErrors
  - WdsPktStatisticsElmnts, [1062](#)
- pRXPacketOverflows
  - WdsPktStatisticsElmnts, [1063](#)
- pRXPacketSuccesses
  - WdsPktStatisticsElmnts, [1063](#)
- pRXStaticGain
  - RXAGCList, [667](#)
- pRXTotalBytes
  - WdsByteTotalsElmnts, [1052](#)
- pRadioInterface
  - nasNetworkTime, [483](#)
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, [908](#)
- pRankIndicatorInd
  - NasSwlIndReg, [497](#)
- pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, [544](#)
- pRatHandoverStatusConfig
  - IMSAIndRegisterInfo, [287](#)
- pRawHorConfidence
  - LocInjectPositionReq, [354](#)
- pRawHorUncCircular
  - LocInjectPositionReq, [354](#)
- pReadAcknowledgementReq
  - cdmaMsgDecodingParams, [146](#)
- pReadResult
  - UIMReadTransparentResp, [821](#)
  - unpack\_uim\_ReadTransparent\_t, [951](#)
- pReason
  - voiceSUPSInfo, [1029](#)
- pRecurrenceType
  - LOCStartReq, [358](#)
  - pack\_loc\_Start\_t, [532](#)
- pRedirNumInfo
  - voiceInfoRec, [1011](#)
- pRefData
  - FirmwareUpdatStat, [222](#)
- pRefString
  - FirmwareUpdatStat, [222](#)
- pRefStringLength
  - FirmwareUpdatStat, [222](#)
- pReferenceNum
  - wcdmaLongMsgDecodingParams, [1035](#)
- pRefreshEvent
  - UIMRefreshGetLastEventResp, [824](#)
- pRegCallStatInfoEvt
  - \_getIndicationRegResp, [56](#)
  - \_setIndicationRegReq, [64](#)
- pRegDTMFEvents
  - voiceIndicationRegisterInfo, [1009](#)
- pRegInd
  - \_getTransLayerInfoResp, [58](#)
- pRegMgrConfigEvents
  - imsCfgIndRegisterInfo, [295](#)
- pRegStatus
  - \_getTransNWRegInfoResp, [59](#)
- pRegStatusConfig
  - IMSAIndRegisterInfo, [287](#)
- pRegStatusErrorCode
  - imsaRegStatusInfo, [290](#)
- pRegTransLayerInfoEvt
  - \_getIndicationRegResp, [56](#)
  - \_setIndicationRegReq, [64](#)
- pRegTransNWRegInfoEvt
  - \_getIndicationRegResp, [56](#)
  - \_setIndicationRegReq, [64](#)
- pRegVoicePrivacyEvents
  - voiceIndicationRegisterInfo, [1009](#)
- pRejectReason
  - pack\_swioa\_SLQSOMADMSendSelection\_t, [558](#)
- pRelValidity
  - cdmaMsgEncodingParams, [147](#)

- pRelativeValidity
  - cdmaMsgDecodingParams, [146](#)
- pRemainingRetries
  - UIMDepersonalizationResp, [813](#)
  - UIMPinResp, [818](#)
  - unpack\_uim\_ChangePin\_t, [949](#)
  - unpack\_uim\_SetPinProtection\_t, [951](#)
  - unpack\_uim\_UnblockPin\_t, [954](#)
  - unpack\_uim\_VerifyPin\_t, [955](#)
- pRemotePartyName
  - voiceCallInfoResp, [985](#)
- pRemotePartyNum
  - voiceCallInfoResp, [985](#)
- pReportChannelRate
  - getDUNCallInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
- pReportConnStatus
  - getDUNCallInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
- pReportDataBearerTech
  - getDUNCallInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
- pReportDormStatus
  - getDUNCallInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
- pReqFieldsList
  - IMSASupportedFieldsResp, [293](#)
- pReqMitigationLvl
  - TmdGetMitigationLvlResp, [786](#)
- pReqSettings
  - pack\_wds\_SLQSGetRuntimeSettings\_t, [574](#)
- pRequestOptionList
  - unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [971](#)
  - WdsDHCPv4Config, [1056](#)
- pRequestedTag
  - pack\_sms\_SLQSGetSMSList\_t, [554](#)
- pRespData
  - USSDRspFNetwork, [977](#)
- pRespFieldsList
  - IMSASupportedFieldsResp, [293](#)
- pRetryCount
  - \_SLQSOMADMSessionInfo, [70](#)
- pRetryMessage
  - slqssendasyncsmsparams\_s, [721](#)
- pRetryMessageId
  - slqssendasyncsmsparams\_s, [721](#)
- pRevInUse
  - CDMASysInfo, [153](#)
  - nas\_CDMASysInfo, [393](#)
- pRevInUseValid
  - CDMASysInfo, [153](#)
  - nas\_CDMASysInfo, [393](#)
- pRevTunneling
  - pack\_wds\_SetMobileIPProfile\_t, [570](#)
- pRingBackTimer
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pRingingTimer
  - GetIMSVoIPConfigResp, [249](#)
  - imsVoIPConfigInfo, [301](#)
  - SetIMSVoIPConfigReq, [693](#)
- pRoamPref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [90](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [905](#)
- pRoamTimer
  - voiceGetConfigReq, [1006](#)
- pRoamTimerCnt
  - voiceGetConfigResp, [1008](#)
- pRoamTimerConfig
  - voiceSetConfigReq, [1021](#)
- pRoamTimerStatus
  - voiceSetConfigResp, [1022](#)
- pRoaming
  - QmiNas3GppNetworkInfo, [642](#)
- pRscp
  - nasSigInfo, [496](#)
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [909](#)
- pRxFilter
  - swiQosModifyReq, [771](#)
  - swiQosReq, [772](#)
- pRxFlow
  - swiQosGranted, [770](#)
  - swiQosModifyReq, [771](#)
  - swiQosReq, [772](#)
- pRxQFilter
  - QosFlowInfo, [651](#)
- pRxQFlowGranted
  - QosFlowInfo, [651](#)
- PSDetachReq, [614](#)
  - pDetachAction, [614](#)
- PSDomain
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- pSIPConfigEvents
  - imsCfgIndRegisterInfo, [295](#)
- pSIPLocalPort
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pSMSAttemptedFlag
  - getDyingGaspStatistics, [243](#)
  - packgetDyingGaspStatistics, [580](#)
- pSMSCAddressInfo
  - SMSEventInfo\_s, [738](#)
- pSMSCConfigEvents
  - imsCfgIndRegisterInfo, [295](#)
- pSMSFormat
  - GetIMSSMSConfigParams, [246](#)
  - imsSMSConfigInfo, [298](#)
  - SetIMSSMSConfigReq, [690](#)
- pSMSOnIMSInfo
  - SMSEventInfo\_s, [738](#)
- pSMSOverIPNwInd

- GetIMSSMSConfigParams, 246
- imsSMSConfigInfo, 298
- SetIMSSMSConfigReq, 690
- pSMSSupport
  - custFeaturesInfo, 181
- pSMSSvcRAT
  - imsaSvcStatusInfo, 294
- pSMSSvcStatus
  - imsaSvcStatusInfo, 294
- pSV
  - BdsSVInfo, 120
  - loc\_BdsSVInfo, 337
  - loc\_SVInfo, 345
  - SVInfo, 755
- pSVInfo
  - LocDelAssDataReq, 348
  - pack\_loc\_Delete\_Assist\_Data\_t, 527
- pSWVerString
  - \_SLQSSwiGetHostDevInfoParams, 75
  - \_SLQSSwiSetHostDevInfoParams, 78
- pSatelliteInfo
  - gnssSvInfoNotification, 260
- pScAddr
  - wcdmaLongMsgDecodingParams, 1035
  - wcdmaMsgDecodingParams, 1036
- pScAddrLength
  - wcdmaLongMsgDecodingParams, 1035
  - wcdmaMsgDecodingParams, 1036
- pScanResult
  - \_slqsNetworkScanInfo, 68
  - unpack\_nas\_PerformNetworkScan\_t, 890
- pScrAmrEnable
  - GetIMSVoIPConfigResp, 249
  - imsVoIPConfigInfo, 301
  - SetIMSVoIPConfigReq, 693
- pScrAmrWbEnable
  - GetIMSVoIPConfigResp, 249
  - imsVoIPConfigInfo, 301
  - SetIMSVoIPConfigReq, 693
- pSeDNSIPv4Address
  - swiPDPRuntimeSettingsResp, 764
- pSeDNSIPv6Address
  - swiPDPRuntimeSettingsResp, 764
- pSePCSCFIPv4Address
  - swiPDPRuntimeSettingsResp, 764
- pSePCSCFIPv6Address
  - swiPDPRuntimeSettingsResp, 764
- pSecDNSIPv4AddPref
  - LibpackProfile3GPP, 313
  - LibPackprofile\_3GPP, 324
  - Profile3GPP, 606
- pSecDNSIPv6addpref
  - LibpackProfile3GPP, 313
  - LibPackprofile\_3GPP, 324
  - Profile3GPP, 606
- pSecV6DnsAddress
  - LibpackProfile3GPP2, 318
  - LibPackprofile\_3GPP2, 329
- Profile3GPP2, 611
- pSecondaryDNSV4
  - qmiWdsRunTimeSettings, 648
- pSecondaryDNSV6
  - qmiWdsRunTimeSettings, 648
- pSecondaryFlag
  - LibpackProfile3GPP, 313
  - LibPackprofile\_3GPP, 324
  - Profile3GPP, 606
- pSecondaryHA
  - pack\_wds\_SetMobileIPProfile\_t, 570
- pSecondaryV4DnsAddress
  - LibpackProfile3GPP2, 318
  - LibPackprofile\_3GPP2, 329
  - Profile3GPP2, 611
- pSectorID
  - NetworkStatEVDO, 508
- pSenderAddr
  - cdmaMsgDecodingParams, 146
  - wcdmaLongMsgDecodingParams, 1035
  - wcdmaMsgDecodingParams, 1036
- pSenderAddrLength
  - cdmaMsgDecodingParams, 146
  - wcdmaLongMsgDecodingParams, 1035
  - wcdmaMsgDecodingParams, 1036
- pSensorDataUsage
  - QmiCbkLocBestAvailPosInd, 625
  - QmiCbkLocPositionReportInd, 635
  - unpack\_loc\_BestAvailPos\_Ind\_t, 874
  - unpack\_loc\_PositionRpt\_Ind\_t, 881
- pServerAddrList
  - qmiWdsRunTimeSettings, 648
- pServiceClass
  - voiceSetSUPSServiceReq, 1025
- pServiceOption
  - slqssendasyncsmsparams\_s, 721
- pServiceStatusConfig
  - IMSAIndRegisterInfo, 287
- pServingSystemInd
  - nasIndicationRegisterReq, 480
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 542
- pSessionExpiryTimer
  - GetIMSVoIPConfigResp, 249
  - imsVoIPConfigInfo, 301
  - SetIMSVoIPConfigReq, 693
- pSessionIDv4
  - GetSessionIDResp, 256
- pSessionIDv6
  - GetSessionIDResp, 256
- pSessionState
  - \_SLQSOMADMSessionInfo, 70
- pSessionType
  - \_SLQSOMADMSessionInfo, 70
- pSettingResp
  - GetIMSSMSConfigParams, 246
  - GetIMSUserConfigParams, 247
  - GetIMSVoIPConfigResp, 249
  - GetRegMgrConfigParams, 256

- GetSIPConfigResp, [257](#)
- SetIMSMSConfigResp, [690](#)
- SetIMSUserConfigResp, [691](#)
- SetIMSVoIPConfigResp, [694](#)
- SetRegMgrConfigResp, [700](#)
- SetSIPConfigResp, [707](#)
- pSeverity
  - \_SLQSOMADMSessionInfo, [70](#)
- pSigCompEnabled
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pSigIndReq
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t, [544](#)
- pSignalInfo
  - voiceInfoRec, [1012](#)
- pSignalStrengthInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pSmsOnlms
  - slqssendasyncsmsparams\_s, [721](#)
  - slqssendsmsparams\_s, [722](#)
- pSmsServiceRat
  - IMSAServiceStatus, [292](#)
- pSmsServiceStatus
  - IMSAServiceStatus, [292](#)
- pSource
  - \_SLQSOMADMSessionInfo, [70](#)
- pSourceIP
  - LibPackTFTIDParams, [332](#)
  - TFTIDParams, [782](#)
- pSourceLength
  - \_SLQSOMADMSessionInfo, [70](#)
- pSpeedHorizontal
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pSpeedUnc
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pSpeedVertical
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pSpeedVerticalUnc
  - QmiCbkLocBestAvailPosInd, [625](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
- pSrcRAT
  - imsaRatStatusInfo, [289](#)
- pSrvDomainPref
  - \_sysSelectPrefInfo, [84](#)
  - \_sysSelectPrefParams, [91](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
  - unpack\_nas\_SLQSSetSysSelectionPref\_t, [905](#)
- pSrvOpt
  - voiceCallInfoResp, [985](#)
- pSrvRegRestriction
  - \_sysSelectPrefParams, [91](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
- pSrvProviderName
  - nasOperatorNameResp, [484](#)
- pStage0Val
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStage1Val
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStage2Val
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStage3Val
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStage4Val
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStageCnt
  - RXPCMIIRFiltr, [670](#)
  - TXPCMIIRFiltr, [795](#)
- pStatMask
  - WdsPktStatisticsReq, [1063](#)
- pStatus
  - \_SLQSOMADMSessionInfo, [70](#)
- pSubnetMaskV4
  - qmiWdsRunTimeSettings, [648](#)
- pSubscribeTimer
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pSubscriptionInfoInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pSupUSBComps
  - USBCompParams, [976](#)
- pSupportedMsgList
  - IMSASupportedMsgInfo, [293](#)
- pSuppsNotifEvents
  - voiceIndicationRegisterInfo, [1009](#)
- pSvUsedforFix
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pSvcClass
  - voiceGetCallBarringReq, [994](#)
  - voiceGetCallBarringResp, [995](#)
  - voiceGetCallFWReq, [996](#)
  - voiceGetCallWaitInfo, [999](#)
  - voiceSUPSInfo, [1029](#)
- pSvcType
  - voiceCallRequestParams, [987](#)

- pSwiGetResetInd
  - dmsIndicationRegisterReq, [205](#)
- pSysInfoInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [542](#)
- pSysInfoNoChange
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pSystemSelectionInd
  - nasIndicationRegisterReq, [480](#)
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [543](#)
- pTCPDstPort
  - swiQosFilter, [766](#)
- pTCPSrcPort
  - swiQosFilter, [766](#)
- pTDSCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMAECIOTresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMAECIOTreshList
  - nas\_TDSCDMAECIOTresh, [453](#)
  - TDSCDMAECIOTresh, [776](#)
- pTDSCDMARSCPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMARSCPTresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMARSCPTreshList
  - nas\_TDSCDMARSCPTresh, [453](#)
  - TDSCDMARSCPTresh, [777](#)
- pTDSCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMARSSITresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [539](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMARSSITreshList
  - nas\_TDSCDMARSSITresh, [454](#)
  - TDSCDMARSSITresh, [777](#)
- pTDSCDMASINRCONFThresh
  - sigInfo, [709](#)
- pTDSCDMASINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMASINRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pTDSCDMASINRThreshList
  - nas\_TDSCDMASINRThresh, [454](#)
  - TDSCDMASINRThresh, [780](#)
- pTDSCDMASigInfoExt
  - nasGetSigInfoResp, [473](#)
  - nasSigInfo, [496](#)
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [909](#)
- pTDSCDMASigInfoRscp
  - nasGetSigInfoResp, [473](#)
- pTFTID1Params
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pTFTID2Params
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pTTYConfigStatus
  - voiceSetConfigResp, [1022](#)
- pTTYMode
  - voiceGetConfigReq, [1007](#)
  - voiceSetConfigReq, [1021](#)
- pTXAGCList
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pTXAIG
  - TXAGCList, [793](#)
- pTXAVCSwitch
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pTXComprSlope
  - TXAGCList, [793](#)
- pTXComprThres
  - TXAGCList, [793](#)
- pTXDroppedCount
  - WdsPktStatisticsElmnts, [1063](#)
- pTXExpSlope
  - TXAGCList, [793](#)
- pTXExpThres
  - TXAGCList, [793](#)
- pTXGain
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pTXInfo
  - nasGetTxRxInfoResp, [477](#)
- pTXOKBytesCount
  - getDUNCallInfoResp, [242](#)
- pTXOKBytesLastCall
  - WdsPktStatisticsElmnts, [1063](#)
- pTXOkBytesCount
  - WdsPktStatisticsElmnts, [1063](#)
- pTXPCMIIRFitr
  - GetAudioPathConfigResp, [233](#)
  - SetAudioPathConfigReq, [685](#)
- pTXPacketErrors
  - WdsPktStatisticsElmnts, [1063](#)
- pTXPacketOverflows
  - WdsPktStatisticsElmnts, [1063](#)
- pTXPacketSuccesses
  - WdsPktStatisticsElmnts, [1063](#)
- pTXStaticGain
  - TXAGCList, [793](#)
- pTXTotalBytes
  - WdsByteTotalsElmnts, [1052](#)
- pTdsBandCapability
  - BandCapabilityResp, [119](#)

- pTdsdmaBandPref
  - \_sysSelectPrefParams, [91](#)
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [549](#)
- pTech
  - pack\_wds\_SLQSStartDataSession\_t, [578](#)
- pTechnology
  - qmiWdsRunTimeSettings, [648](#)
  - ssdatasession\_params, [751](#)
- pTechnologyMask
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pTextMsg
  - cdmaMsgDecodingParams, [146](#)
  - cdmaMsgEncodingParams, [148](#)
  - wcdmaLongMsgDecodingParams, [1035](#)
  - wcdmaMsgDecodingParams, [1036](#)
  - wcdmaMsgEncodingParams, [1037](#)
- pTextMsgLength
  - cdmaMsgDecodingParams, [146](#)
  - wcdmaLongMsgDecodingParams, [1035](#)
  - wcdmaMsgDecodingParams, [1036](#)
- pTgtRAT
  - imsaRatStatusInfo, [289](#)
- pTime
  - \_SLQSOMADMSessionInfo, [70](#)
  - SwiOTAMsg\_s, [761](#)
- pTimeLength
  - \_SLQSOMADMSessionInfo, [70](#)
- pTimeSrc
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pTimeStamp
  - getDyingGaspStatistics, [243](#)
  - packgetDyingGaspStatistics, [580](#)
  - PDSPositionData, [584](#)
- pTimeType
  - PDSPositionData, [584](#)
- pTimeUnc
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pTimeZone
  - nasNetworkTime, [483](#)
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, [908](#)
- pTimerSIPReg
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pTimerT1
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pTimerT2
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pTimerTf
  - GetSIPConfigResp, [257](#)
  - imsSIPConfigInfo, [297](#)
  - SetSIPConfigReq, [706](#)
- pTimerVal
  - voiceSetSUPSServiceReq, [1025](#)
- pTimestampAge
  - LocInjectPositionReq, [354](#)
- pTimestampUtc
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pTokenBucket
  - swiQosFlow, [770](#)
- pTos
  - swiQosFilter, [766](#)
- pTotalBytesRX
  - QosEventInfo, [650](#)
  - slqsWdsEventInfo, [731](#)
- pTotalBytesTX
  - QosEventInfo, [650](#)
  - slqsWdsEventInfo, [731](#)
- pTotalNum
  - wcdmaLongMsgDecodingParams, [1035](#)
- pTrafficClass
  - swiQosFlow, [770](#)
- pTranDstPort
  - swiQosFilter, [766](#)
- pTranSrcPort
  - swiQosFilter, [766](#)
- pTransLayerInfo
  - \_getTransLayerInfoResp, [58](#)
  - \_transLayerInfoNotification, [92](#)
- pTransferRouteMTMessageInfo
  - SMSEventInfo\_s, [738](#)
- pTransferStatInd
  - getDUNCaIInfoReq, [239](#)
  - pack\_wds\_SLQSGetDUNCaIInfo\_t, [572](#)
  - wdsSetEventReportReq, [1067](#)
- pTransferStatusReport
  - smsSetRoutesReq, [744](#)
- pTrueIMSI
  - nasGet3GPP2SubscriptionInfoResp, [470](#)
- pTxFilter
  - swiQosModifyReq, [771](#)
  - swiQosReq, [772](#)
- pTxFlow
  - swiQosGranted, [770](#)
  - swiQosModifyReq, [771](#)
  - swiQosReq, [772](#)
- pTxQFilter
  - QosFlowInfo, [651](#)

- pTxQFlowGranted
  - QosFlowInfo, [651](#)
- pTypeCode
  - USSDRespFNetwork, [977](#)
- pUATI
  - GetHRPDStatsResp, [245](#)
- pUDPDstPort
  - swiQosFilter, [766](#)
- pUDPSrcPort
  - swiQosFilter, [766](#)
- pUMTSCellID
  - nasCellLocationInfoResp, [469](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [906](#)
- pUMTSGrantedQoS
  - qmiWdsRunTimeSettings, [648](#)
- pUMTSInfo
  - nasCellLocationInfoResp, [469](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [907](#)
- pUMTSMinQoS
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pUMTSMinQoSSigInd
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pUMTSReqQoS
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pUMTSReqQoSSigInd
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pUSBComp
  - USBCompConfig, [974](#)
  - USBCompParams, [976](#)
- pUSSDData
  - USSDNoWaitIndicationInfo, [977](#)
- pUSSDInfo
  - USSResp, [979](#)
- pUSSInfo
  - voiceSUPSInfo, [1029](#)
- pUTSvcRAT
  - imsaSvcStatusInfo, [294](#)
- pUTSvcStatus
  - imsaSvcStatusInfo, [294](#)
- pUUSInFo
  - voiceCallRequestParams, [987](#)
- pUUSInfo
  - voiceCallInfoResp, [985](#)
- pUimSlotsStatus
  - UIMGetSlotsStatusResp, [817](#)
  - unpack\_uim\_SLQSUIIMGetSlotsStatus\_t, [953](#)
- pUpdateCompleteStatus
  - \_SLQSOMADMSessionInfo, [71](#)
- pUser
  - pack\_wds\_SLQSSStartDataSession\_t, [579](#)
- pUserAcknowledgementReq
  - cdmaMsgDecodingParams, [146](#)
- pUserConfigEvents
  - imsCfgIndRegisterInfo, [295](#)
- pUserData
  - slqssendasyncsmsparams\_s, [721](#)
- pUserId
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pUserIdSize
  - LibpackProfile3GPP2, [318](#)
  - LibPackprofile\_3GPP2, [329](#)
  - Profile3GPP2, [611](#)
- pUsername
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
  - Profile3GPP, [606](#)
  - qmiWdsRunTimeSettings, [648](#)
  - ssdatasession\_params, [752](#)
- pUsernameSize
  - LibpackProfile3GPP, [313](#)
  - LibPackprofile\_3GPP, [324](#)
  - Profile3GPP, [606](#)
- pUtServiceRat
  - IMSAServiceStatus, [292](#)
- pUtServiceStatus
  - IMSAServiceStatus, [292](#)
- pV4sessionId
  - WdsByteTotals, [1051](#)
  - WdsConnectionRate, [1053](#)
  - WdsPktStatisticsResp, [1064](#)
- pV6sessionId
  - WdsByteTotals, [1051](#)
  - WdsConnectionRate, [1053](#)
  - WdsPktStatisticsResp, [1064](#)
- pVOIPSvcRAT
  - imsaSvcStatusInfo, [294](#)
- pVOIPSvcStatus
  - imsaSvcStatusInfo, [294](#)
- pVTSvcRAT
  - imsaSvcStatusInfo, [294](#)
- pVTSvcStatus
  - imsaSvcStatusInfo, [294](#)
- pVerboseFailReasonType
  - unpack\_wds\_SLQSSStartDataSession\_t, [971](#)
- pVerboseFailureReason
  - unpack\_wds\_SLQSSStartDataSession\_t, [971](#)
- pVersionString
  - \_SLQSSwiGetOSInfoParams, [76](#)
  - \_SLQSSwiSetOSInfoParams, [79](#)
- pVertConfidence
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [635](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)

- pVertReliability
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [636](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pVertUnc
  - LocInjectPositionReq, [354](#)
  - QmiCbkLocBestAvailPosInd, [625](#)
  - QmiCbkLocPositionReportInd, [636](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
- pVerticalConfidence
  - PDSPositionData, [584](#)
- pVerticalUnc
  - PDSPositionData, [584](#)
- pVoIPConfigEvents
  - imsCfgIndRegisterInfo, [296](#)
- pVoiceDomainPref
  - voiceGetConfigReq, [1007](#)
- pVoiceDomainPrefStatus
  - voiceSetConfigResp, [1022](#)
- pVoicePrivacy
  - voiceCallInfoResp, [986](#)
  - voiceGetAllCallInfo, [993](#)
- pVoipServiceRat
  - IMSAServiceStatus, [292](#)
- pVoipServiceStatus
  - IMSAServiceStatus, [292](#)
- pVolume
  - SetM2MAudioProfileReq, [696](#)
- pVsServiceRat
  - IMSAServiceStatus, [292](#)
- pVsServiceStatus
  - IMSAServiceStatus, [292](#)
- pVtServiceRat
  - IMSAServiceStatus, [292](#)
- pVtServiceStatus
  - IMSAServiceStatus, [292](#)
- pWCDMABER
  - GetErrRateResp, [244](#)
- pWCDMACallBarringSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pWCDMACipherDomainSysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pWCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pWCDMAECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pWCDMAECIOThreshList
  - nas\_WCDMAECIOThresh, [462](#)
  - WCDMAECIOThresh, [1032](#)
- pWCDMAInfoLTNeighborCell
  - nasCellLocationInfoResp, [469](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [907](#)
- pWCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pWCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [540](#)
  - setSignalStrengthInfo, [705](#)
- pWCDMARSSIThreshList
  - nas\_WCDMARSSIThresh, [464](#)
  - WCDMARSSIThresh, [1038](#)
- pWCDMASSInfo
  - nasGetSigInfoResp, [473](#)
- pWCDMASigInfo
  - nasSigInfo, [496](#)
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [909](#)
- pWCDMASrvStatusInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pWCDMASysInfo
  - nasGetSysInfoResp, [476](#)
  - nasSysInfo, [501](#)
  - unpack\_nas\_SLQSGetSysInfo\_t, [901](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [914](#)
- pWcdmaUARFCN
  - nasSwiGetChannelLockResp, [496](#)
  - nasSwiSetChannelLockReq, [498](#)
- pWifiState
  - PDSPosMethodStateReq, [585](#)
- pXid
  - QmiCbkLocBestAvailPosInd, [625](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
- pXtraDataState
  - PDSPosMethodStateReq, [585](#)
- pXtraTimeState
  - PDSPosMethodStateReq, [585](#)
- pack\_dms\_GetActivationState
  - dms.h, [1079](#)
- pack\_dms\_GetBandCapability
  - dms.h, [1079](#)
- pack\_dms\_GetCrashAction
  - dms.h, [1079](#)
- pack\_dms\_GetCustFeature
  - dms.h, [1080](#)
- pack\_dms\_GetCustFeaturesV2
  - dms.h, [1080](#)
- pack\_dms\_GetCustFeaturesV2\_t, [520](#)
  - cust\_id, [521](#)
  - list\_type, [521](#)
  - Tlvresult, [521](#)
- pack\_dms\_GetDeviceCap
  - dms.h, [1080](#)
- pack\_dms\_GetDeviceCapabilities

- dms.h, [1080](#)
- pack\_dms\_GetDeviceHardwareRev
  - dms.h, [1081](#)
- pack\_dms\_GetDeviceMfr
  - dms.h, [1081](#)
- pack\_dms\_GetDeviceSerialNumbers
  - dms.h, [1081](#)
- pack\_dms\_GetFSN
  - dms.h, [1083](#)
- pack\_dms\_GetFirmwareInfo
  - dms.h, [1082](#)
- pack\_dms\_GetFirmwareRevision
  - dms.h, [1082](#)
- pack\_dms\_GetFirmwareRevisions
  - dms.h, [1082](#)
- pack\_dms\_GetHardwareRevision
  - dms.h, [1083](#)
- pack\_dms\_GetIMSI
  - dms.h, [1083](#)
- pack\_dms\_GetModelID
  - dms.h, [1084](#)
- pack\_dms\_GetNetworkTime
  - dms.h, [1084](#)
- pack\_dms\_GetPRLVersion
  - dms.h, [1085](#)
- pack\_dms\_GetPower
  - dms.h, [1084](#)
- pack\_dms\_GetSerialNumbers
  - dms.h, [1085](#)
- pack\_dms\_GetUSBComp
  - dms.h, [1086](#)
- pack\_dms\_GetVoiceNumber
  - dms.h, [1086](#)
- pack\_dms\_SLQSDmsSwiGetResetInfo
  - dms.h, [1089](#)
- pack\_dms\_SLQSDmsSwiIndicationRegister
  - dms.h, [1089](#)
- pack\_dms\_SLQSDmsSwiIndicationRegister\_t, [523](#)
  - resetInfoInd, [524](#)
- pack\_dms\_SLQSGetBandCapability
  - dms.h, [1090](#)
- pack\_dms\_SLQSSwiClearDyingGaspStatistics
  - dms.h, [1090](#)
- pack\_dms\_SLQSSwiGetDyingGaspCfg
  - dms.h, [1091](#)
- pack\_dms\_SLQSSwiGetDyingGaspStatistics
  - dms.h, [1091](#)
- pack\_dms\_SLQSSwiGetFirmwareCurr
  - dms.h, [1091](#)
- pack\_dms\_SLQSSwiGetFwUpdateStatus
  - dms.h, [1092](#)
- pack\_dms\_SLQSSwiSetDyingGaspCfg
  - dms.h, [1092](#)
- pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [524](#)
  - pDestSMSContent, [524](#)
  - pDestSMSNum, [524](#)
- pack\_dms\_SetCrashAction
  - dms.h, [1086](#)
- pack\_dms\_SetCrashAction\_t, [521](#)
  - crashAction, [521](#)
- pack\_dms\_SetCustFeature
  - dms.h, [1087](#)
- pack\_dms\_SetCustFeature\_t, [521](#)
  - DHCPRelayEnabled, [522](#)
  - DisableIMSI, [522](#)
  - GPSPMP, [522](#)
  - GPSSel, [522](#)
  - GpsEnable, [522](#)
  - IPFamSupport, [522](#)
  - IsVoiceEnabled, [522](#)
  - RMAutoConnect, [522](#)
  - SMSSupport, [522](#)
- pack\_dms\_SetCustFeaturesV2
  - dms.h, [1087](#)
- pack\_dms\_SetCustFeaturesV2\_t, [522](#)
  - cust\_id, [522](#)
  - cust\_value, [522](#)
  - Tlvresult, [522](#)
  - value\_length, [523](#)
- pack\_dms\_SetEventReport
  - dms.h, [1088](#)
- pack\_dms\_SetEventReport\_t, [523](#)
  - mode, [523](#)
- pack\_dms\_SetFirmwarePreference
  - dms.h, [1088](#)
- pack\_dms\_SetPower
  - dms.h, [1088](#)
- pack\_dms\_SetPower\_t, [523](#)
  - mode, [523](#)
  - Tlvresult, [523](#)
- pack\_dms\_SetUSBComp
  - dms.h, [1089](#)
- pack\_dms\_SetUSBComp\_t, [523](#)
  - Tlvresult, [523](#)
  - USBComp, [523](#)
- pack\_dms\_UIMGetICCID
  - dms.h, [1092](#)
- pack\_dms\_UIMGetICCID\_t, [524](#)
  - Tlvresult, [525](#)
- pack\_fms\_GetImagesPreference
  - fms.h, [1109](#)
- pack\_fms\_GetImagesPreference\_t, [525](#)
  - Tlvresult, [525](#)
- pack\_fms\_GetStoredImages
  - fms.h, [1109](#)
- pack\_fms\_GetStoredImages\_t, [525](#)
  - Tlvresult, [525](#)
- pack\_fms\_SetImagesPreference
  - fms.h, [1109](#)
- pack\_fms\_SetImagesPreference\_t, [525](#)
  - bForceDownload, [526](#)
  - imageListSize, [526](#)
  - modemindex, [526](#)
  - plmageList, [526](#)
  - Tlvresult, [526](#)
- pack\_loc\_Delete\_Assist\_Data\_t, [526](#)

- pBdsSVInfo, [527](#)
  - pCellDb, [527](#)
  - pClkInfo, [527](#)
  - pGnssData, [527](#)
  - pSVInfo, [527](#)
  - Tlvresult, [527](#)
- pack\_loc\_DeleteAssistData
  - loc.h, [1115](#)
- pack\_loc\_EventRegister
  - loc.h, [1115](#)
- pack\_loc\_EventRegister\_t, [527](#)
  - eventRegister, [529](#)
  - Tlvresult, [529](#)
- pack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, [1116](#)
- pack\_loc\_SLQSLOCGetBestAvailPos\_t, [530](#)
  - Tlvresult, [530](#)
  - xid, [530](#)
- pack\_loc\_SetExtPowerState
  - loc.h, [1115](#)
- pack\_loc\_SetExtPowerState\_t, [529](#)
  - extPowerState, [529](#)
  - Tlvresult, [529](#)
- pack\_loc\_SetOperationMode
  - loc.h, [1116](#)
- pack\_loc\_SetOperationMode\_t, [530](#)
  - mode, [530](#)
  - Tlvresult, [530](#)
- pack\_loc\_Start
  - loc.h, [1116](#)
- pack\_loc\_Start\_t, [530](#)
  - pApplicationInfo, [532](#)
  - pConfigAltitudeAssumed, [532](#)
  - pHorizontalAccuracyLvl, [532](#)
  - pIntermediateReportState, [532](#)
  - pMinIntervalTime, [532](#)
  - pRecurrenceType, [532](#)
  - SessionId, [532](#)
  - Tlvresult, [532](#)
- pack\_loc\_Stop
  - loc.h, [1117](#)
- pack\_loc\_Stop\_t, [532](#)
  - SessionId, [532](#)
  - Tlvresult, [532](#)
- pack\_nas\_GetACCOLC
  - nas.h, [1127](#)
- pack\_nas\_GetANAAAAAuthenticationStatus
  - nas.h, [1128](#)
- pack\_nas\_GetCDMANetworkParameters
  - nas.h, [1128](#)
- pack\_nas\_GetHomeNetwork
  - nas.h, [1128](#)
- pack\_nas\_GetNetworkPreference
  - nas.h, [1129](#)
- pack\_nas\_GetRFInfo
  - nas.h, [1129](#)
- pack\_nas\_GetServingNetwork
  - nas.h, [1129](#)
- pack\_nas\_GetServingNetworkCapabilities
  - nas.h, [1129](#)
- pack\_nas\_GetSignalStrengths
  - nas.h, [1130](#)
- pack\_nas\_PerformNetworkScan
  - nas.h, [1130](#)
- pack\_nas\_SLQSGetNetworkTime
  - nas.h, [1132](#)
- pack\_nas\_SLQSGetPLMNName
  - nas.h, [1132](#)
- pack\_nas\_SLQSGetPLMNName\_t, [534](#)
  - mcc, [534](#)
  - mnc, [534](#)
  - pMncPcsStatus, [534](#)
- pack\_nas\_SLQSGetServingSystem
  - nas.h, [1132](#)
- pack\_nas\_SLQSGetSignalStrength
  - nas.h, [1132](#)
- pack\_nas\_SLQSGetSysInfo
  - nas.h, [1133](#)
- pack\_nas\_SLQSGetSysSelectionPref
  - nas.h, [1133](#)
- pack\_nas\_SLQSInitiateNetworkRegistration
  - nas.h, [1133](#)
- pack\_nas\_SLQSInitiateNetworkRegistration\_t, [534](#)
  - pChangeDuration, [535](#)
  - pMNRInfo, [535](#)
  - pMncPcsDigitStatus, [535](#)
  - regAction, [535](#)
- pack\_nas\_SLQSNasConfigSigInfo2
  - nas.h, [1134](#)
- pack\_nas\_SLQSNasConfigSigInfo2\_t, [535](#)
  - pHDRIODelta, [539](#)
  - pHDRIOTresh, [539](#)
  - pLTESigRptConfig, [539](#)
- pack\_nas\_SLQSNasGetCellLocationInfo
  - nas.h, [1134](#)
- pack\_nas\_SLQSNasGetSigInfo
  - nas.h, [1134](#)
- pack\_nas\_SLQSNasIndicationRegisterExt
  - nas.h, [1135](#)
- pack\_nas\_SLQSNasIndicationRegisterExt\_t, [540](#)
  - pDDTMInd, [542](#)
  - pDualStandByPrefInd, [542](#)
  - pErrorRateInd, [542](#)
  - pHDRSessionCloseInd, [542](#)
  - pLTECphyCa, [542](#)
  - pManagedRoamingInd, [542](#)
  - pNetworkTimeInd, [542](#)
  - pServingSystemInd, [542](#)
  - pSignalStrengthInd, [542](#)
  - pSubscriptionInfoInd, [542](#)
  - pSysInfoInd, [542](#)
  - pSystemSelectionInd, [543](#)
- pack\_nas\_SLQSNasSwiModemStatus
  - nas.h, [1135](#)
- pack\_nas\_SLQSNasSwiOTAMessageCallback
  - nas.h, [1135](#)

- pack\_nas\_SLQSNasSwiOTAMessageCallback\_t, 543
  - gsmUmtsDI, 543
  - gsmUmtsUI, 544
  - lteEmmDI, 544
  - lteEmmUI, 544
  - lteEsmDI, 544
  - lteEsmUI, 544
  - pRankIndicatorInd, 544
- pack\_nas\_SLQSSetBandPreference
  - nas.h, 1136
- pack\_nas\_SLQSSetSignalStrengthsCallback
  - nas.h, 1136
- pack\_nas\_SLQSSetSignalStrengthsCallback\_t, 544
  - bEnable, 544
  - pSigIndReq, 544
- pack\_nas\_SLQSSetSysSelectionPref
  - nas.h, 1136
- pack\_nas\_SLQSSetSysSelectionPref\_t, 544
  - pAcqOrderPref, 548
  - pBandPref, 548
  - pCSGID, 548
  - pChgDuration, 548
  - pEmerMode, 548
  - pGWAqOrderPref, 548
  - pLTEBandPref, 548
  - pModePref, 549
  - pNetSelPref, 549
  - pPRLPref, 549
  - pRAT, 549
  - pRoamPref, 549
  - pSrvDomainPref, 549
  - pSrvRegRestriction, 549
  - pTdsdmaBandPref, 549
- pack\_nas\_SLQSSwiGetLteCQI
  - nas.h, 1137
- pack\_nas\_SetACCOLC
  - nas.h, 1130
- pack\_nas\_SetACCOLC\_t, 532
  - accolc, 533
  - spc, 533
- pack\_nas\_SetLURejectCallback
  - nas.h, 1131
- pack\_nas\_SetNetworkPreference
  - nas.h, 1131
- pack\_nas\_SetNetworkPreference\_t, 533
  - Duration, 533
  - TechnologyPref, 534
  - Tlvresult, 534
- pack\_nas\_SetRFInfoCallback
  - nas.h, 1131
- pack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, 1131
- pack\_qmi\_t, 549
  - msgid, 549
  - svc, 549
  - timeout, 549
  - xid, 549
- pack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, 1555
- pack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, 1555
- pack\_qos\_SLQSQosSwiReadApnExtraParams\_t, 549
  - apnId, 550
- pack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, 1556
- pack\_qos\_SLQSQosSwiReadDataStats\_t, 550
  - apnId, 550
- pack\_qos\_SLQSSetQosEventCallback
  - qos.h, 1557
- pack\_qos\_SLQSSetQosEventCallback\_t, 550
  - enable, 551
- pack\_sms\_SLQSDeleteSMS
  - sms.h, 1566
- pack\_sms\_SLQSDeleteSMS\_t, 552
  - pMessageIndex, 552
  - pMessageMode, 553
  - pMessageTag, 553
  - storageType, 553
- pack\_sms\_SLQSGetSMS
  - sms.h, 1566
- pack\_sms\_SLQSGetSMS\_t, 553
  - messageIndex, 553
  - pMessageMode, 553
  - storageType, 553
- pack\_sms\_SLQSGetSMSList
  - sms.h, 1566
- pack\_sms\_SLQSGetSMSList\_t, 553
  - pMessageMode, 554
  - pRequestedTag, 554
  - storageType, 554
- pack\_sms\_SLQSModifySMSStatus
  - sms.h, 1567
- pack\_sms\_SLQSModifySMSStatus\_t, 554
  - messageIndex, 555
  - messageTag, 555
  - pMessageMode, 555
  - storageType, 555
- pack\_sms\_SendSMS
  - sms.h, 1565
- pack\_sms\_SendSMS\_t, 551
  - messageFormat, 551
  - messageSize, 551
  - pLinktimer, 551
  - pMessage, 551
- pack\_sms\_SetNewSMSCallback
  - sms.h, 1565
- pack\_sms\_SetNewSMSCallback\_t, 551
  - status, 552
- pack\_swilloc\_SwiLocGetAutoStart
  - swilloc.h, 1572
- pack\_swilloc\_SwiLocSetAutoStart
  - swilloc.h, 1572
- pack\_swilloc\_SwiLocSetAutoStart\_t, 555
  - fix\_rate, 556
  - fix\_type, 556
  - function, 556

- max\_dist, [556](#)
- max\_time, [556](#)
- set\_fix\_rate, [556](#)
- set\_fix\_type, [556](#)
- set\_function, [556](#)
- set\_max\_dist, [556](#)
- set\_max\_time, [556](#)
- pack\_swima\_SLQSOMADMAAlertCallback
  - swioma.h, [1574](#)
- pack\_swima\_SLQSOMADMCancelSession
  - swioma.h, [1575](#)
- pack\_swima\_SLQSOMADMCancelSession\_t, [556](#)
  - sessionType, [557](#)
- pack\_swima\_SLQSOMADMGetSessionInfo
  - swioma.h, [1575](#)
- pack\_swima\_SLQSOMADMGetSessionInfo\_t, [557](#)
  - SessionType, [557](#)
- pack\_swima\_SLQSOMADMGetSettings
  - swioma.h, [1576](#)
- pack\_swima\_SLQSOMADMSendSelection
  - swioma.h, [1576](#)
- pack\_swima\_SLQSOMADMSendSelection\_t, [557](#)
  - pDeferTime, [558](#)
  - pRejectReason, [558](#)
  - selection, [558](#)
- pack\_swima\_SLQSOMADMSetSettings
  - swioma.h, [1577](#)
- pack\_swima\_SLQSOMADMSetSettings\_t, [558](#)
  - FOTAdownload, [559](#)
  - pAutosdm, [559](#)
  - pFwAutoCheck, [559](#)
- pack\_swima\_SLQSOMADMStartSession
  - swioma.h, [1578](#)
- pack\_swima\_SLQSOMADMStartSession\_t, [559](#)
  - sessionType, [559](#)
- pack\_uim\_ChangePin
  - uim.h, [1584](#)
- pack\_uim\_ChangePin\_t, [560](#)
  - changePIN, [560](#)
  - EncryptedPIN1, [560](#)
  - pIndicationToken, [560](#)
  - pKeyReferenceID, [560](#)
  - sessionInfo, [560](#)
  - Tlvresult, [560](#)
- pack\_uim\_GetCardStatus
  - uim.h, [1584](#)
- pack\_uim\_ReadTransparent
  - uim.h, [1585](#)
- pack\_uim\_ReadTransparent\_t, [561](#)
  - fileIndex, [561](#)
  - pEncryptData, [561](#)
  - pIndicationToken, [561](#)
  - readTransparent, [561](#)
  - sessionInfo, [562](#)
  - Tlvresult, [562](#)
- pack\_uim\_SLQSUIEventRegister
  - uim.h, [1585](#)
- pack\_uim\_SLQSUIEventRegister\_t, [563](#)
  - eventMask, [563](#)
- pack\_uim\_SLQSUIMGetSlotsStatus
  - uim.h, [1586](#)
- pack\_uim\_SLQSUIMSwitchSlot
  - uim.h, [1586](#)
- pack\_uim\_SLQSUIMSwitchSlot\_t, [563](#)
  - bLogicalSlot, [564](#)
  - ulPhysicalSlot, [564](#)
- pack\_uim\_SetPinProtection
  - uim.h, [1585](#)
- pack\_uim\_SetPinProtection\_t, [562](#)
  - EncryptedPIN1, [562](#)
  - pIndicationToken, [562](#)
  - pKeyReferenceID, [563](#)
  - pinProtection, [562](#)
  - sessionInfo, [563](#)
  - Tlvresult, [563](#)
- pack\_uim\_UnblockPin
  - uim.h, [1586](#)
- pack\_uim\_UnblockPin\_t, [564](#)
  - EncryptedPIN1, [565](#)
  - pIndicationToken, [565](#)
  - pKeyReferenceID, [565](#)
  - pinProtection, [565](#)
  - sessionInfo, [565](#)
  - Tlvresult, [565](#)
- pack\_uim\_VerifyPin
  - uim.h, [1587](#)
- pack\_uim\_VerifyPin\_t, [565](#)
  - pEncryptedPIN1, [566](#)
  - pIndicationToken, [566](#)
  - pKeyReferenceID, [566](#)
  - sessionInfo, [566](#)
  - Tlvresult, [566](#)
  - verifyPIN, [566](#)
- pack\_wds\_GetConnectionRate
  - wds.h, [1595](#)
- pack\_wds\_GetDefaultProfile
  - wds.h, [1596](#)
- pack\_wds\_GetDefaultProfile\_t, [566](#)
  - profiletype, [566](#)
- pack\_wds\_GetDefaultProfileNum
  - wds.h, [1596](#)
- pack\_wds\_GetDefaultProfileNum\_t, [566](#)
  - family, [567](#)
  - type, [567](#)
- pack\_wds\_GetDormancyState
  - wds.h, [1597](#)
- pack\_wds\_GetDormancyState\_t, [567](#)
- pack\_wds\_GetLastMobileIPError
  - wds.h, [1597](#)
- pack\_wds\_GetLastMobileIPError\_t, [567](#)
- pack\_wds\_GetMobileIP
  - wds.h, [1597](#)
- pack\_wds\_GetMobileIP\_t, [567](#)
- pack\_wds\_GetMobileIPProfile
  - wds.h, [1598](#)
- pack\_wds\_GetMobileIPProfile\_t, [567](#)

- index, [567](#)
- pack\_wds\_GetPacketStatus
  - wds.h, [1598](#)
- pack\_wds\_GetPacketStatus\_t, [567](#)
  - statmask, [568](#)
- pack\_wds\_GetSessionDuration
  - wds.h, [1599](#)
- pack\_wds\_GetSessionDuration\_t, [568](#)
- pack\_wds\_GetSessionState
  - wds.h, [1599](#)
- pack\_wds\_RMSetTransferStatistics
  - wds.h, [1600](#)
- pack\_wds\_RMSetTransferStatistics\_t, [568](#)
  - RmTrasnferStaticsReq, [568](#)
- pack\_wds\_SLQSCreateProfile
  - wds.h, [1601](#)
- pack\_wds\_SLQSCreateProfile\_t, [571](#)
  - pCurProfile, [571](#)
  - pProfileId, [571](#)
  - pProfileType, [571](#)
- pack\_wds\_SLQSDeleteProfile
  - wds.h, [1601](#)
- pack\_wds\_SLQSDeleteProfile\_t, [571](#)
  - profileIndex, [572](#)
  - profileType, [572](#)
- pack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, [1602](#)
- pack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, [1602](#)
- pack\_wds\_SLQSGetCurrDataSystemStat\_t, [572](#)
- pack\_wds\_SLQSGetDUNCallInfo
  - wds.h, [1603](#)
- pack\_wds\_SLQSGetDUNCallInfo\_t, [572](#)
  - Mask, [572](#)
  - pReportChannelRate, [572](#)
  - pReportConnStatus, [572](#)
  - pReportDataBearerTech, [572](#)
  - pReportDormStatus, [572](#)
  - pTransferStatInd, [572](#)
- pack\_wds\_SLQSGetDataBearerTechnology
  - wds.h, [1603](#)
- pack\_wds\_SLQSGetDataBearerTechnology\_t, [572](#)
- pack\_wds\_SLQSGetProfileSettings
  - wds.h, [1604](#)
- pack\_wds\_SLQSGetProfileSettings\_t, [573](#)
  - ProfileId, [573](#)
  - ProfileType, [573](#)
- pack\_wds\_SLQSGetRuntimeSettings
  - wds.h, [1604](#)
- pack\_wds\_SLQSGetRuntimeSettings\_t, [573](#)
  - pReqSettings, [574](#)
- pack\_wds\_SLQSModifyProfile
  - wds.h, [1604](#)
- pack\_wds\_SLQSModifyProfile\_t, [574](#)
  - curProfile, [575](#)
  - pProfileId, [575](#)
  - pProfileType, [575](#)
- pack\_wds\_SLQSSetDHCPv4ClientConfig
  - wds.h, [1606](#)
- pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [577](#)
  - pProfileId, [577](#)
- pack\_wds\_SLQSSet3GPPConfigItem
  - wds.h, [1605](#)
- pack\_wds\_SLQSSet3GPPConfigItem\_t, [575](#)
  - \_3gppRelease, [576](#)
  - defaultPDNEnabled, [576](#)
  - profileList, [576](#)
- pack\_wds\_SLQSSetIPFamilyPreference
  - wds.h, [1605](#)
- pack\_wds\_SLQSSetIPFamilyPreference\_t, [576](#)
  - IPFamilyPreference, [576](#)
- pack\_wds\_SLQSSetWdsEventCallback
  - wds.h, [1606](#)
- pack\_wds\_SLQSSetWdsEventCallback\_t, [576](#)
  - currentDataBearer, [577](#)
  - dataBearer, [577](#)
  - dataSystemStatus, [577](#)
  - dormancyStatus, [577](#)
  - interval, [577](#)
  - mobileIP, [577](#)
  - transferStats, [577](#)
- pack\_wds\_SLQSStartDataSession
  - wds.h, [1606](#)
- pack\_wds\_SLQSStartDataSession\_t, [577](#)
  - pAuth, [578](#)
  - pPass, [578](#)
  - pTech, [578](#)
  - pUser, [579](#)
  - pprofileid3gpp, [578](#)
  - pprofileid3gpp2, [578](#)
- pack\_wds\_SLQSStopDataSession
  - wds.h, [1607](#)
- pack\_wds\_SLQSStopDataSession\_t, [579](#)
  - psid, [579](#)
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings
  - wds.h, [1607](#)
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [579](#)
  - contextId, [579](#)
  - contextType, [579](#)
- pack\_wds\_SetDefaultProfile
  - wds.h, [1600](#)
- pack\_wds\_SetDefaultProfile\_t, [568](#)
  - authentication, [569](#)
  - ipAddress, [569](#)
  - pApnname, [569](#)
  - pName, [569](#)
  - pPassword, [569](#)
  - pUsername, [569](#)
  - pdpType, [569](#)
  - primaryDNS, [569](#)
  - profileType, [569](#)
  - secondaryDNS, [569](#)
- pack\_wds\_SetDefaultProfileNum
  - wds.h, [1600](#)
- pack\_wds\_SetDefaultProfileNum\_t, [569](#)
  - family, [569](#)

- index, [569](#)
- type, [569](#)
- pack\_wds\_SetMobileIPProfile
  - wds.h, [1601](#)
- pack\_wds\_SetMobileIPProfile\_t, [569](#)
  - index, [570](#)
  - pAAASPI, [570](#)
  - pAddress, [570](#)
  - pEnabled, [570](#)
  - pHASPI, [570](#)
  - pMNAAS, [570](#)
  - pMNHA, [570](#)
  - pNAI, [570](#)
  - pPrimaryHA, [570](#)
  - pRevTunneling, [570](#)
  - pSecondaryHA, [570](#)
  - spc, [571](#)
- PackCreateProfileOut, [579](#)
  - ExtErrorCode, [579](#)
  - ProfileIndex, [579](#)
  - ProfileType, [580](#)
- package\_name
  - omaDmFotaTlv, [516](#)
  - omaDmFotaTlvExt, [518](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- packageSize
  - omaDmFotaTlvExt, [518](#)
- packageid\_str
  - slqsfwinfo\_s, [717](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [849](#)
- packetSrvStatus
  - qaGobiApiCbk.h, [1169](#)
- packetZone
  - CDMASysInfo, [153](#)
  - nas\_CDMASysInfo, [393](#)
- packetZoneValid
  - CDMASysInfo, [153](#)
  - nas\_CDMASysInfo, [393](#)
- packgetDyingGaspCfg, [580](#)
  - pDestSMSContent, [580](#)
  - pDestSMSNum, [580](#)
- packgetDyingGaspStatistics, [580](#)
  - pSMSAttemptedFlag, [580](#)
  - pTimeStamp, [580](#)
- path
  - fileInfo, [220](#)
  - uim\_fileInfo, [802](#)
- pathLen
  - fileInfo, [220](#)
  - uim\_fileInfo, [802](#)
- pbIMSRegistered
  - imsaRegStatusInfo, [290](#)
- pci
  - cellParams, [155](#)
  - ltePCI, [368](#)
  - nas\_cellParams, [395](#)
  - nas\_PhyCaAggPcellInfo, [434](#)
  - nas\_PhyCaAggScellIndType, [436](#)
  - nas\_PhyCaAggScellInfo, [439](#)
  - nas\_umtsLTENbrCell, [460](#)
  - NASPhyCaAggPcellInfo, [485](#)
  - NASPhyCaAggScellIndType, [487](#)
  - NASPhyCaAggScellInfo, [488](#)
  - PhyCaAggPcellInfo, [589](#)
  - PhyCaAggScellIndType, [591](#)
  - PhyCaAggScellInfo, [593](#)
  - umtsLTENbrCell, [835](#)
- pcsfFQDNAddress
  - PCSCFFQDNAddressList, [582](#)
  - wds\_PCSCFFQDNAddressList, [1047](#)
- pdpType
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
- pdptype
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- peakRate
  - tokenBucket, [788](#)
  - unpack\_qos\_tokenBucket\_t, [936](#)
- peakThroughputClass
  - GPRSQoS, [261](#)
  - GPRSRequestedQoS, [262](#)
  - LibPackGPRSRequestedQoS, [307](#)
  - wds\_GPRSQoS, [1045](#)
- peerNumberInfo, [585](#)
  - callID, [587](#)
  - numLen, [587](#)
  - numPI, [587](#)
  - numPlan, [587](#)
  - numSI, [587](#)
  - numType, [587](#)
  - number, [587](#)
- PerformNetworkScan
  - qaGobiApiNas.h, [1344](#)
- PersistentTechPref
  - unpack\_nas\_GetNetworkPreference\_t, [887](#)
- persoFeature
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- persoRetries
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- persoState
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- persoUnblockRetries
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- personalizationStatus, [587](#)
  - feature, [588](#)
  - numFeatures, [588](#)
  - unblockLeft, [588](#)
  - verifyLeft, [588](#)
- pfailureCause

- USSResp, 978
- phase
  - rxInfo, 668
- PhyCaAggPcellInfo, 588
  - dl\_bw\_value, 589
  - freq, 589
  - iLTEbandValue, 589
  - NasGetLTECphyCaInfo, 472
  - pci, 589
  - TlvPresent, 589
- PhyCaAggScellIDIBw, 589
  - dl\_bw\_value, 589
  - NasGetLTECphyCaInfo, 472
  - TlvPresent, 590
- PhyCaAggScellIndType, 590
  - freq, 591
  - NasGetLTECphyCaInfo, 472
  - pci, 591
  - scell\_state, 591
  - TlvPresent, 591
- PhyCaAggScellIndex, 590
  - NasGetLTECphyCaInfo, 472
  - scell\_idx, 590
  - TlvPresent, 590
- PhyCaAggScellInfo, 591
  - dl\_bw\_value, 593
  - freq, 593
  - iLTEbandValue, 593
  - NasGetLTECphyCaInfo, 472
  - pci, 593
  - scell\_state, 593
  - TlvPresent, 593
- PhysicalLayer
  - protocolSubtypeElement, 613
- PilotEnergy
  - NetworkStatEVDO, 508
- PilotPN
  - PilotSetParams, 594
- PilotSetData, 593
  - NumPilots, 593
  - pPilotSetInfo, 593
- PilotSetParams, 594
  - PilotPN, 594
  - PilotStrength, 594
  - PilotType, 594
- PilotStrength
  - PilotSetParams, 594
- PilotType
  - PilotSetParams, 594
- pin1Len
  - encryptedPIN1, 213
  - uim\_encryptedPIN1, 801
- pin1Retries
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- pin1State
  - appStats, 103
- appStatus, 106
  - uim\_appStatus, 798
- pin1Val
  - encryptedPIN1, 213
  - uim\_encryptedPIN1, 801
- pin2Retries
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- pin2State
  - appStats, 103
  - appStatus, 106
  - uim\_appStatus, 798
- pinID
  - changeUIMPIN, 156
  - setPINProtection, 699
  - uim\_changeUIMPIN, 801
  - uim\_setPINProtection, 806
  - uim\_unblockUIMPIN, 809
  - uim\_verifyUIMPIN, 810
  - unblockUIMPIN, 842
  - verifyUIMPIN, 980
- pinLen
  - changeUIMPIN, 156
  - uim\_changeUIMPIN, 801
  - uim\_verifyUIMPIN, 810
  - verifyUIMPIN, 980
- pinLength
  - setPINProtection, 699
  - uim\_setPINProtection, 806
- pinOperation
  - setPINProtection, 699
  - uim\_setPINProtection, 806
- pinProtection
  - pack\_uim\_SetPinProtection\_t, 562
  - pack\_uim\_UnblockPin\_t, 565
  - UIMSetPinProtectionReq, 827
- pinVal
  - changeUIMPIN, 156
  - uim\_changeUIMPIN, 801
  - uim\_verifyUIMPIN, 810
  - verifyUIMPIN, 980
- pinValue
  - setPINProtection, 699
  - uim\_setPINProtection, 806
- PkQmiNasGetRFBandInfo
  - qaNasGetRFBandInfo.h, 1545
- PkQmiNasPerformNetworkScan
  - qaNasPerformNetworkScan.h, 1546
- PkgDescLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- PkgDescription
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- PkgName
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946

- PkgNameLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- pkgver
  - CurrentImgList, 174
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 864
- PktErrRate
  - unpack\_qos\_swIQosFlow\_t, 936
- pktErrRate, 594
  - exponent, 594
  - multiplier, 594
- PktStatElmntsV4
  - WdsPktStatisticsResp, 1064
- PktStatElmntsV6
  - WdsPktStatisticsResp, 1064
- plmn
  - GERANInfo, 228
  - LTEInfoIntrafreq, 365
  - nas\_GERANInfo, 404
  - nas\_LTEInfoIntrafreq, 422
  - nas\_UMTSInfo, 458
  - UMTSInfo, 833
- polarityIncluded
  - lineCtrlInfo, 336
- Port, 597
  - port, 597
  - range, 597
- port
  - Port, 597
  - unpack\_qos\_Port\_t, 921
- Position Determination Service (PDS), 37
- pprofileid3gpp
  - pack\_wds\_SLQSStartDataSession\_t, 578
- pprofileid3gpp2
  - pack\_wds\_SLQSStartDataSession\_t, 578
- prDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 972
- prDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 972
- prPCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 973
- prPCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 973
- Precedence
  - unpack\_qos\_swIQosFilter\_t, 932
- precedenceClass
  - GPRSQoS, 261
  - GPRSRequestedQoS, 262
  - LibPackGPRSRequestedQoS, 307
  - wds\_GPRSQoS, 1045
- precisionDilution
  - qaGobiApiCbK.h, 1170
- precisionDilution\_s, 597
  - HDOP, 598
- PDOP, 598
- VDOP, 598
- PrefImageList, 598
  - listEntries, 598
  - listSize, 598
- prefNetwork
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 963
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- prefVoiceSO, 598
  - evrcCapability, 600
  - homeOrigVoiceSO, 600
  - homePageVoiceSO, 600
  - namID, 600
  - roamOrigVoiceSO, 600
- Preferred
  - nas\_QmiNas3GppNetworkInfo, 440
  - SlqsNas3GppNetworkInfo, 718
- prefixLen
  - IPv6Addr, 304
  - unpack\_qos\_IPv6Addr\_t, 919
- presentationInd
  - ECTNum, 212
  - remotePartyNum, 659
- priChA
  - CDMAChannel, 141
- priChB
  - CDMAChannel, 141
- priSize
  - unpack\_dms\_GetFirmwareRevisions\_t, 850
- pridns
  - unpack\_wds\_GetDefaultProfile\_t, 956
- pridnsv6
  - unpack\_wds\_GetDefaultProfile\_t, 956
- primaryDNS
  - pack\_wds\_SetDefaultProfile\_t, 569
- PrimaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 967
- PrimaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 967
- primaryHA
  - unpack\_wds\_GetMobileIPProfile\_t, 959
- privacyPref
  - voiceSetPrefPrivacy, 1023
- priver
  - CurrentImgList, 174
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 864
- priversion\_str
  - slqsfwinfo\_s, 717
  - unpack\_dms\_GetFirmwareInfo\_t, 849
- Profile
  - GetAudioPathConfigReq, 231
  - GetAudioProfileResp, 234
  - GetAudioVoITLBCConfigReq, 235
  - GetM2MAudioProfileResp, 251
  - GetM2MAudioVolumeReq, 252
  - GetM2MAVMuteReq, 252

- GetM2MSpkrGainReq, [254](#)
- SetAudioPathConfigReq, [685](#)
- SetAudioProfileReq, [686](#)
- SetAudioVolTLBConfigReq, [687](#)
- SetM2MAudioAVCFGRReq, [695](#)
- SetM2MAudioProfileReq, [696](#)
- SetM2MAudioVolumeReq, [697](#)
- SetM2MAVMuteReq, [698](#)
- SetM2MSpkrGainReq, [698](#)
- Profile3GPP, [600](#)
  - pAPNClass, [605](#)
  - pAPNDisabledFlag, [605](#)
  - pAPNName, [605](#)
  - pAPNnameSize, [605](#)
  - pAddrAllocPref, [605](#)
  - pAuthenticationPref, [605](#)
  - pGPRSMinimumQoS, [605](#)
  - pGPRSRequestedQos, [605](#)
  - pIPv4AddrPref, [605](#)
  - pIPv6AddPref, [605](#)
  - plmCnFlag, [605](#)
  - pPDNInactivTimeout, [605](#)
  - pPDPtype, [606](#)
  - pPassword, [605](#)
  - pPasswordSize, [605](#)
  - pPcscfAddrUsingDhcp, [605](#)
  - pPcscfAddrUsingPCO, [605](#)
  - pPdpAccessConFlag, [605](#)
  - pPdpContext, [605](#)
  - pPdpDataCompType, [605](#)
  - pPdpHdrCompType, [606](#)
  - pPriDNSIPv4AddPref, [606](#)
  - pPriDNSIPv6addpref, [606](#)
  - pPrimaryID, [606](#)
  - pProfilename, [606](#)
  - pProfilenameSize, [606](#)
  - pQosClassID, [606](#)
  - pSecDNSIPv4AddPref, [606](#)
  - pSecDNSIPv6addpref, [606](#)
  - pSecondaryFlag, [606](#)
  - pTFTID1Params, [606](#)
  - pTFTID2Params, [606](#)
  - pUMTSMinQoS, [606](#)
  - pUMTSMinQoSSigInd, [606](#)
  - pUMTSReqQoS, [606](#)
  - pUMTSReqQoSSigInd, [606](#)
  - pUsername, [606](#)
  - pUsernameSize, [606](#)
- Profile3GPP2, [606](#)
  - pAPNClass3GPP2, [610](#)
  - pAPNEnabled3GPP2, [610](#)
  - pAllowLinger, [610](#)
  - pApnString, [610](#)
  - pApnStringSize, [610](#)
  - pAppPriority, [610](#)
  - pAppType, [610](#)
  - pAuthPassword, [610](#)
  - pAuthPasswordSize, [611](#)
  - pAuthProtocol, [611](#)
  - pAuthRetryCount, [611](#)
  - pAuthTimeout, [611](#)
  - pDataMode, [611](#)
  - pDataRate, [611](#)
  - plpcpAckTimeout, [611](#)
  - plpcpCreqRetryCount, [611](#)
  - plsPcscfAddressNedded, [611](#)
  - pLcpAckTimeout, [611](#)
  - pLcpCreqRetryCount, [611](#)
  - pNegoDnsSrvrPref, [611](#)
  - pPDNInactivTimeout3GPP2, [611](#)
  - pPdnType, [611](#)
  - pPppSessCloseTimer1x, [611](#)
  - pPppSessCloseTimerDO, [611](#)
  - pPriV6DnsAddress, [611](#)
  - pPrimaryV4DnsAddress, [611](#)
  - pRATType, [611](#)
  - pSecV6DnsAddress, [611](#)
  - pSecondaryV4DnsAddress, [611](#)
  - pUserId, [611](#)
  - pUserIdSize, [611](#)
- ProfileID
  - \_GetProfileSettingIn, [56](#)
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- ProfileId
  - pack\_wds\_SLQSGetProfileSettings\_t, [573](#)
- profileId
  - WdsDHCPv4ProfileId, [1059](#)
  - wdsDhcpv4ProfileId, [1060](#)
- ProfileId3GPP2
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- ProfileIdentifier, [611](#)
  - profileIndex, [612](#)
  - profileType, [612](#)
- ProfileIndex
  - PackCreateProfileOut, [579](#)
- profileIndex
  - pack\_wds\_SLQSDeleteProfile\_t, [572](#)
  - ProfileIdentifier, [612](#)
  - SLQSDeleteProfileParams, [716](#)
  - wds\_ProfileIdentifier, [1048](#)
- profileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [576](#)
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, [963](#)
- ProfileName
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- ProfileType
  - \_GetProfileSettingIn, [56](#)
  - pack\_wds\_SLQSGetProfileSettings\_t, [573](#)
  - PackCreateProfileOut, [580](#)
  - unpack\_wds\_SLQSGetProfileSettings\_t, [965](#)
- profileType
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
  - pack\_wds\_SLQSDeleteProfile\_t, [572](#)
  - ProfileIdentifier, [612](#)
  - SLQSDeleteProfileParams, [716](#)
  - wds\_ProfileIdentifier, [1048](#)

- WdsDHCPv4ProfileId, [1060](#)
- wdsDhcpv4ProfileId, [1060](#)
- profiletype
  - pack\_wds\_GetDefaultProfile\_t, [566](#)
- Protocol
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- protocolSubtypeElement, [612](#)
  - AccessMac, [613](#)
  - AuthProt, [613](#)
  - ControlMac, [613](#)
  - EncryptProt, [613](#)
  - ForwardMac, [613](#)
  - IdleState, [613](#)
  - KeyExchange, [613](#)
  - MultDisc, [613](#)
  - PhysicalLayer, [613](#)
  - ReverseMac, [613](#)
  - SecProt, [613](#)
  - VirtStream, [614](#)
- ProvisionStatus
  - CLIPResp, [158](#)
  - CLIRResp, [158](#)
  - CNAPResp, [160](#)
  - COLPResp, [161](#)
  - COLRResp, [162](#)
- psAttachState
  - nas\_servSystem, [446](#)
  - NASServingSystemInfo, [494](#)
  - ServingSystemInfo, [680](#)
  - servSystem, [681](#)
- psBarStatus
  - CallBarringSysInfo, [122](#)
  - callBarStatus, [123](#)
  - nas\_CallBarringSysInfo, [387](#)
  - nas\_callBarStatus, [388](#)
- psState
  - CommInfo, [163](#)
  - nas\_CommInfo, [397](#)
- psc
  - nas\_UMTSInfo, [458](#)
  - nas\_wcdmaCellInfo, [462](#)
  - nas\_WCDMASysInfo, [467](#)
  - UMTSInfo, [834](#)
  - wcdmaCellInfo, [1031](#)
  - WCDMASysInfo, [1041](#)
- pscValid
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1041](#)
- pscsfIPv4Addr
  - PCSCFIPv4ServerAddressList, [583](#)
  - wds\_PCSCFIPv4ServerAddressList, [1047](#)
- psid
  - pack\_wds\_SLQSSStopDataSession\_t, [579](#)
  - unpack\_wds\_SLQSSStartDataSession\_t, [971](#)
- puk1Retries
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- puk2Retries
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- pukLen
  - uim\_unblockUIMPIN, [809](#)
  - unblockUIMPIN, [842](#)
- pukVal
  - uim\_unblockUIMPIN, [809](#)
  - unblockUIMPIN, [842](#)
- pv4sessionId
  - WdsIpAddressInfoReq, [1061](#)
- pv6sessionId
  - WdsIpAddressInfoReq, [1061](#)
- pwrDenialTime
  - lineCtrlInfo, [336](#)
- QMI\_SAR\_RF\_STATE\_1
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_2
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_3
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_4
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_5
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_6
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_7
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_8
  - qaGobiApiSar.h, [1389](#)
- QMI\_SAR\_RF\_STATE\_DEFAULT
  - qaGobiApiSar.h, [1389](#)
- QMI\_WDS\_CURRENT\_CALL\_DB\_MASK
  - qaGobiApiWds.h, [1509](#)
- QMI\_WDS\_LAST\_CALL\_DB\_MASK
  - qaGobiApiWds.h, [1509](#)
- QCI
  - LibPackQosClassID, [330](#)
  - QosClassID, [649](#)
- QCWWAN2kConnect
  - qaGobiApiDcs.h, [1238](#)
- QCWWAN2kEnumerateDevices
  - qaGobiApiDcs.h, [1238](#)
- QCWWAN2kGetConnectedDeviceID
  - qaGobiApiDcs.h, [1239](#)
- QCWWANConnect
  - qaGobiApiDcs.h, [1239](#)
- QCWWANDisconnect
  - qaGobiApiDcs.h, [1240](#)
- QCWWANEnumerateDevices
  - qaGobiApiDcs.h, [1240](#)
- QFlowState
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- QLIC
  - DeviceConfigDetail, [198](#)
- qaCbkCatEventReportInd.h

- eTLV\_CBK\_ALPHA\_IDENTIFIER, 1150
- eTLV\_CBK\_DISPLAY\_TEXT, 1150
- eTLV\_CBK\_END\_PROACTIVE\_SESSION, 1150
- eTLV\_CBK\_GET\_IN\_KEY, 1150
- eTLV\_CBK\_GET\_INPUT, 1150
- eTLV\_CBK\_LANGUAGE\_NOTIFICATION, 1150
- eTLV\_CBK\_REFRESH, 1150
- eTLV\_CBK\_SELECT\_ITEM, 1150
- eTLV\_CBK\_SETUP\_EVENT\_LIST, 1150
- eTLV\_CBK\_SETUP\_IDLE\_MODE\_TEXT, 1150
- eTLV\_CBK\_SETUP\_MENU, 1150
- eTLV\_END\_PROACTIVE\_SESSION\_LENGTH, 1150
- eTLV\_REFRESH\_LENGTH, 1150
- eTLV\_SETUP\_EVENT\_LIST\_LENGTH, 1150
- qaCbkSwiOmaDmEventReportInd.h
  - eTLV\_IND\_OMA\_DM\_CONFIG, 1151
  - eTLV\_IND\_OMA\_DM\_FOTA, 1151
  - eTLV\_IND\_OMA\_DM\_NOT, 1151
- qaGobiApiCbk.h
  - DEVICE\_STATE\_BOOT, 1200
  - DEVICE\_STATE\_DISCONNECTED, 1200
  - DEVICE\_STATE\_READY, 1200
  - eQA\_QMI\_SVC\_NA, 1200
  - eQA\_QMI\_SVC\_NAS, 1200
  - eQA\_QMI\_SVC\_WDS, 1200
  - SMS\_EVENT\_ETWS, 1200
  - SMS\_EVENT\_ETWS\_PLMN, 1200
  - SMS\_EVENT\_MESSAGE\_MODE, 1200
  - SMS\_EVENT\_MT\_MESSAGE, 1200
  - SMS\_EVENT\_SMS\_ON\_IMS, 1200
  - SMS\_EVENT\_SMSC\_ADDRESS, 1200
  - SMS\_EVENT\_TRANSFER\_ROUTE\_MT\_MESSAGE, 1200
- qaGobiApiFms.h
  - eGOBI\_DEV\_SERIES\_9X15, 1287
  - eGOBI\_DEV\_SERIES\_9X30, 1287
  - eGOBI\_DEV\_SERIES\_G3K, 1287
  - eGOBI\_DEV\_SERIES\_NON\_GOBI, 1287
  - eGOBI\_DEV\_SERIES\_SIERRA\_GOBI, 1287
  - eGOBI\_DEV\_SERIES\_UNKNOWN, 1287
  - eGOBI\_IMG\_CAR\_3, 1288
  - eGOBI\_IMG\_CAR\_AERIS, 1289
  - eGOBI\_IMG\_CAR\_ALLTEL, 1288
  - eGOBI\_IMG\_CAR\_AMX\_TELCEL, 1289
  - eGOBI\_IMG\_CAR\_ATT, 1288
  - eGOBI\_IMG\_CAR\_BELL, 1288
  - eGOBI\_IMG\_CAR\_BHARTI, 1288
  - eGOBI\_IMG\_CAR\_BRASIL\_VIVO, 1289
  - eGOBI\_IMG\_CAR\_CHINA\_MOBILE, 1288
  - eGOBI\_IMG\_CAR\_CHINA\_TELECOM, 1288
  - eGOBI\_IMG\_CAR\_CHINA\_UNICOM, 1288
  - eGOBI\_IMG\_CAR\_EMOBILE, 1288
  - eGOBI\_IMG\_CAR\_FACTORY, 1288
  - eGOBI\_IMG\_CAR\_GENERIC, 1288
  - eGOBI\_IMG\_CAR\_GENERIC\_CDMA, 1288
  - eGOBI\_IMG\_CAR\_IUSACELL, 1288
  - eGOBI\_IMG\_CAR\_KDDI, 1288
  - eGOBI\_IMG\_CAR\_KT\_FREETEL, 1288
  - eGOBI\_IMG\_CAR\_LEAP, 1288
  - eGOBI\_IMG\_CAR\_METROPCS, 1288
  - eGOBI\_IMG\_CAR\_NETCOM, 1289
  - eGOBI\_IMG\_CAR\_NORF, 1288
  - eGOBI\_IMG\_CAR\_NTT\_DOCOMO, 1288
  - eGOBI\_IMG\_CAR\_O2, 1288
  - eGOBI\_IMG\_CAR\_OMH, 1288
  - eGOBI\_IMG\_CAR\_ORANGE, 1288
  - eGOBI\_IMG\_CAR\_RELIANCE1, 1288
  - eGOBI\_IMG\_CAR\_RELIANCE2, 1288
  - eGOBI\_IMG\_CAR\_ROGERS, 1289
  - eGOBI\_IMG\_CAR\_SFR, 1288
  - eGOBI\_IMG\_CAR\_SINGTEL\_OPTUS, 1288
  - eGOBI\_IMG\_CAR\_SK\_TELCOM1, 1288
  - eGOBI\_IMG\_CAR\_SK\_TELCOM2, 1288
  - eGOBI\_IMG\_CAR\_SOFTBANK, 1288
  - eGOBI\_IMG\_CAR\_SPRINT, 1288
  - eGOBI\_IMG\_CAR\_SWISSCOM, 1288
  - eGOBI\_IMG\_CAR\_TATA, 1288
  - eGOBI\_IMG\_CAR\_TELCOM\_ITALIA, 1288
  - eGOBI\_IMG\_CAR\_TELCOM\_NZ, 1288
  - eGOBI\_IMG\_CAR\_TELEFONICA, 1288
  - eGOBI\_IMG\_CAR\_TELNOR, 1289
  - eGOBI\_IMG\_CAR\_TELIASONERA, 1289
  - eGOBI\_IMG\_CAR\_TELSTRA1, 1288
  - eGOBI\_IMG\_CAR\_TELSTRA2, 1288
  - eGOBI\_IMG\_CAR\_TELUS, 1288
  - eGOBI\_IMG\_CAR\_TMOBILE, 1288
  - eGOBI\_IMG\_CAR\_US, 1288
  - eGOBI\_IMG\_CAR\_VERIZON, 1288
  - eGOBI\_IMG\_CAR\_VODAFONE, 1288
  - eGOBI\_IMG\_GPS\_ASSISTED, 1289
  - eGOBI\_IMG\_GPS\_NO\_XTRA, 1289
  - eGOBI\_IMG\_GPS\_NONE, 1289
  - eGOBI\_IMG\_GPS\_STAND\_ALONE, 1289
  - eGOBI\_IMG\_REG\_ASIA, 1289
  - eGOBI\_IMG\_REG\_AUS, 1289
  - eGOBI\_IMG\_REG\_EU, 1289
  - eGOBI\_IMG\_REG\_GLOBAL, 1289
  - eGOBI\_IMG\_REG\_LA, 1289
  - eGOBI\_IMG\_REG\_NA, 1289
  - eGOBI\_IMG\_TECH\_CDMA, 1289
  - eGOBI\_IMG\_TECH\_UMTS, 1289
  - eGobi\_DEV\_SERIES\_MC83, 1287
- qaGobiApiNas.h
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_100, 1334
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_15, 1334
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_25, 1334
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_50, 1334
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_6, 1334
  - eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_75, 1334
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED, 1334
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED, 1334
  - eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED, 1334

- eNAS\_RADIO\_IF\_GSM, [1333](#)
- eNAS\_RADIO\_IF\_LTE, [1333](#)
- eNAS\_RADIO\_IF\_TDSCDMA, [1333](#)
- eNAS\_RADIO\_IF\_UMTS, [1333](#)
- eSYS\_SRV\_DOMAIN\_CAMPED, [1333](#)
- eSYS\_SRV\_DOMAIN\_CS\_ONLY, [1333](#)
- eSYS\_SRV\_DOMAIN\_CS\_PS, [1333](#)
- eSYS\_SRV\_DOMAIN\_NO\_SRV, [1333](#)
- eSYS\_SRV\_DOMAIN\_PS\_ONLY, [1333](#)
- eSYS\_SRV\_DOMAIN\_UNKNOWN, [1333](#)
- qaGobiApiPds.h
  - eSetServiceAutomaticTrackingDisable, [1367](#)
  - eSetServiceAutomaticTrackingEnable, [1367](#)
- qaGobiApiSar.h
  - QMI\_SAR\_RF\_STATE\_1, [1389](#)
  - QMI\_SAR\_RF\_STATE\_2, [1389](#)
  - QMI\_SAR\_RF\_STATE\_3, [1389](#)
  - QMI\_SAR\_RF\_STATE\_4, [1389](#)
  - QMI\_SAR\_RF\_STATE\_5, [1389](#)
  - QMI\_SAR\_RF\_STATE\_6, [1389](#)
  - QMI\_SAR\_RF\_STATE\_7, [1389](#)
  - QMI\_SAR\_RF\_STATE\_8, [1389](#)
  - QMI\_SAR\_RF\_STATE\_DEFAULT, [1389](#)
- qaGobiApiVoice.h
  - VOICE\_SUPS\_SRV\_CLASS\_DATA, [1485](#)
  - VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITASYNC, [1486](#)
  - VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITSYN-C, [1486](#)
  - VOICE\_SUPS\_SRV\_CLASS\_FAX, [1485](#)
  - VOICE\_SUPS\_SRV\_CLASS\_NONE, [1485](#)
  - VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS, [1486](#)
  - VOICE\_SUPS\_SRV\_CLASS\_PADACCESS, [1486](#)
  - VOICE\_SUPS\_SRV\_CLASS\_SMS, [1486](#)
  - VOICE\_SUPS\_SRV\_CLASS\_VOICE, [1485](#)
- qaGobiApiWds.h
  - QMI\_WDS\_CURRENT\_CALL\_DB\_MASK, [1509](#)
  - QMI\_WDS\_LAST\_CALL\_DB\_MASK, [1509](#)
- qaNasGetRFBandInfo.h
  - eTLV\_RF\_BAND\_INFO, [1545](#)
- qaNasPerformNetworkScan.h
  - eTLV\_3GPP\_NETWORK\_INFO, [1546](#)
- qaCbkCatEventReportInd.h, [1149](#)
  - UpkQmiCbkCatEventReportInd, [1150](#)
- qaCbkSwiOmaDmEventReportInd.h, [1150](#)
  - UpkQmiCbkSwiOmaDmEventReportInd, [1151](#)
  - UpkQmiCbkSwiOmaDmEventReportIndExt, [1151](#)
- qaGobiApiAudio.h, [1151](#)
  - SLQSGetAudioPathConfig, [1152](#)
  - SLQSGetAudioProfile, [1152](#)
  - SLQSGetAudioVoITLBConfig, [1153](#)
  - SLQSSetAudioPathConfig, [1153](#)
  - SLQSSetAudioProfile, [1154](#)
  - SLQSSetAudioVoITLBConfig, [1154](#)
- qaGobiApiCat.h, [1155](#)
  - CATSendEnvelopeCommand, [1155](#)
  - CATSendTerminalResponse, [1156](#)
- qaGobiApiCbk.h, [1156](#)
  - accelAcceptReady, [1166](#)
  - accelTempAcceptReady, [1166](#)
  - CBK\_ENABLE\_EVENT, [1164](#)
  - CBK\_NOCHANGE, [1164](#)
  - DEREGISTER\_EVENT, [1164](#)
  - DEREGISTER\_SRV, [1164](#)
  - device\_state\_enum, [1200](#)
  - eDevState, [1166](#)
  - eQaQMIService, [1200](#)
  - eSMSEventType, [1167](#)
  - EVENT\_MASK\_CARD, [1164](#)
  - FIRST\_INSTANCE, [1164](#)
  - gpsTime, [1167](#)
  - gyroAcceptReady, [1167](#)
  - gyroTempAcceptReady, [1167](#)
  - INVALID\_INSTACNE, [1164](#)
  - IPV4, [1164](#)
  - IPV4V6, [1164](#)
  - IPV6, [1164](#)
  - iSLQSSetDUNCallInfoCallback, [1201](#)
  - iSLQSSetSignalStrengthsCallback, [1201](#)
  - iSLQSSetWdsFirstInstEventCallback, [1201](#)
  - iSLQSSetWdsSecondInstEventCallback, [1201](#)
  - iSLQSSetWdsThirdInstEventCallback, [1201](#)
  - iSLQSSetWdsXferStatsFirstInstCallback, [1201](#)
  - iSLQSSetWdsXferStatsSecondInstCallback, [1201](#)
  - iSetCATEventCallback, [1201](#)
  - iSetSignalStrengthCallback, [1201](#)
  - LteNasReleaseInfo, [1168](#)
  - MAX\_NO\_OF\_CALLS, [1165](#)
  - MAX\_NO\_OF\_FILES, [1165](#)
  - MAX\_NO\_OF\_SLOTS, [1165](#)
  - MAX\_PATH\_LENGTH, [1165](#)
  - MAXUSSDLENGTH, [1165](#)
  - modemTempNotification, [1168](#)
  - NAS\_SRV, [1165](#)
  - NUM\_OF\_SET, [1165](#)
  - PDS\_SRV, [1165](#)
  - packetSrvStatus, [1169](#)
  - precisionDilution, [1170](#)
  - REGISTER\_EVENT, [1165](#)
  - REGISTER\_SRV, [1165](#)
  - ResetInfoNotification, [1170](#)
  - SECOND\_INSTANCE, [1165](#)
  - SLQSNasNetworkTimeCallBack, [1215](#)
  - SLQSNasSigInfo2CallBack, [1215](#)
  - SLQSNasSigInfoCallBack, [1216](#)
  - SLQSNasSwiOTAMessageCallback, [1217](#)
  - SLQSNasSysInfoCallBack, [1217](#)
  - SLQSSetBandPreferenceCbk, [1218](#)
  - SLQSSetDHCPv4ClientLeaseStatusCallback, [1218](#)
  - SLQSSetDUNCallInfoCallback, [1219](#)
  - SLQSSetDataSystemStatusCallback, [1218](#)
  - SLQSSetIMSAPdpStatusCallback, [1219](#)
  - SLQSSetIMSARatStatusCallback, [1220](#)
  - SLQSSetIMSARegStatusCallback, [1220](#)

- SLQSSetIMSASvcStatusCallback, 1220
- SLQSSetIMSSMSConfigCallback, 1221
- SLQSSetIMSUserConfigCallback, 1221
- SLQSSetIMSVoIPConfigCallback, 1222
- SLQSSetLocInjectPositionCallback, 1222
- SLQSSetLocInjectUTCTimeCallback, 1222
- SLQSSetModemTempCallback, 1223
- SLQSSetPacketSrvStatusCallback, 1223
- SLQSSetQosEventCallback, 1223
- SLQSSetQosNWStatusCallback, 1224
- SLQSSetQosPriEventCallback, 1224
- SLQSSetQosStatusCallback, 1225
- SLQSSetRegMgrConfigCallback, 1225
- SLQSSetSDKTerminatedCallback, 1226
- SLQSSetSIPConfigCallback, 1228
- SLQSSetSMSEventCallback, 1228
- SLQSSetServingSystemCallback, 1226
- SLQSSetSessionStateCallback, 1227
- SLQSSetSignalStrengthsCallback, 1227
- SLQSSetSwiGetResetInfoCallback, 1228
- SLQSSetSwiHDRPersCallback, 1228
- SLQSSetSysSelectionPrefCallBack, 1229
- SLQSSetTransLayerInfoCallback, 1229
- SLQSSetTransNWRegInfoCallback, 1230
- SLQSSetWdsEventCallback, 1230
- SLQSSetWdsTransferStatisticCallback, 1231
- SLQSTmdMitigationLvlRptCallback, 1232
- SLQSUIMSetRefreshCallBack, 1232
- SLQSUIMSetStatusChangeCallBack, 1232
- SLQSVoiceInfoRecCallback, 1233
- SLQSVoiceSetAllCallStatusCallBack, 1233
- SLQSVoiceSetDTMFEventCallBack, 1233
- SLQSVoiceSetOTASPStatusCallBack, 1234
- SLQSVoiceSetPrivacyChangeCallBack, 1234
- SLQSVoiceSetSUPSCallBack, 1235
- SLQSVoiceSetSUPSNotificationCallback, 1235
- SLQSWmsAsyncRawSendCallBack, 1236
- SLQSWmsMemoryFullCallBack, 1236
- SLQSWmsMessageWaitingCallBack, 1236
- SMSAsyncRawSend, 1172
- SMSCAddressInfo, 1173
- SMSEtwsMessageInfo, 1173
- SMSEtwsPlmnInfo, 1173
- SMSEventInfo, 1173
- SMSEventType, 1200
- SMSMTMessageInfo, 1174
- SMSMessageModelInfo, 1174
- SMSOnIMSInfo, 1175
- SMSTransferRouteMTMessageInfo, 1175
- sensorDataUsage, 1171
- sessionInformation, 1171
- sessionInformationExt, 1172
- SetActivationStatusCallback, 1201
- SetCATEventCallback, 1201
- SetDataCapabilitiesCallback, 1202
- SetDeviceStateChangeCbk, 1203
- SetFwDldCompletionCbk, 1203
- SetGPSCallback, 1204
- SetLURejectCallback, 1207
- SetLocBestAvailPosCallback, 1204
- SetLocCradleMountCallback, 1204
- SetLocDeleteAssistDataCallback, 1205
- SetLocEngineStateCallback, 1205
- SetLocEventPositionCallback, 1205
- SetLocEventTimeSyncCallback, 1205
- SetLocGnssSvInfoCallback, 1206
- SetLocInjectSensorDataCallback, 1206
- SetLocInjectTimeCallback, 1206
- SetLocOpModeCallback, 1207
- SetLocSensorStreamingCallback, 1207
- SetLocSetExtPowerConfigCallback, 1207
- SetMobileIPStatusCallback, 1208
- SetNMEACallback, 1210
- SetNasLTECphyCalIndCallback, 1208
- SetNetChangeCbk, 1209
- SetNewSMSCallback, 1209
- SetOMADMStateCallback, 1210
- SetPDSSStateCallback, 1210
- SetPowerCallback, 1211
- SetRFInfoCallback, 1211
- SetRMTTransferStatisticsCallback, 1211
- SetRankIndicatorCallback, 1211
- SetRoamingIndicatorCallback, 1212
- SetSLQSOMADMAAlertCallback, 1213
- SetSLQSOMADMAAlertCallbackExt, 1213
- SetSignalStrengthCallback, 1212
- SetUSSDNoWaitIndicationCallback, 1214
- SetUSSDNotificationCallback, 1214
- SetUSSDReleaseCallback, 1215
- SetUimSlotStatusChangeCallback, 1214
- svUsedforFix, 1175
- SwiOTAMsg, 1176
- tFNASwiLTECphyCallInfo, 1177
- tFNASwiOTAMsg, 1177
- tFNActivationStatus, 1176
- tFNAllCallStatus, 1177
- tFNAsyncRawSend, 1177
- tFNBandPreference, 1178
- tFNBestAvailPos, 1179
- tFNCATEvent, 1180
- tFNCbkUimSlotStatusChangeInd, 1180
- tFNDHCPv4ClientLeaseStatus, 1181
- tFNDTMFEvent, 1182
- tFNDUNCallInfo, 1182
- tFNDataCapabilities, 1180
- tFNDataSysStatus, 1181
- tFNDeIAssistData, 1181
- tFNDeviceStateChange, 1181
- tFNEventPosition, 1182
- tFNFwDldCompletion, 1182
- tFNGnssSvInfo, 1183
- tFNHDRPersonaity, 1183
- tFNImmsRegMgrConfig, 1184
- tFNImmsSIPConfig, 1184
- tFNImmsSMSCConfig, 1184
- tFNImmsUserConfig, 1184

- tFNImSVoIPConfig, [1185](#)
- tFNImSaPdpStatus, [1183](#)
- tFNImSaRatStatus, [1183](#)
- tFNImSaRegStatus, [1183](#)
- tFNImSaSvcStatus, [1184](#)
- tFNInfoRec, [1185](#)
- tFNInjectPosition, [1185](#)
- tFNInjectSensorData, [1185](#)
- tFNInjectTimeStatus, [1185](#)
- tFNInjectUTCtime, [1185](#)
- tFNLURReject, [1186](#)
- tFNMemoryFull, [1187](#)
- tFNMessageWaiting, [1187](#)
- tFNMitILvlRpt, [1187](#)
- tFNMobileIPStatus, [1187](#)
- tFNModemTempInfo, [1187](#)
- tFNNet, [1187](#)
- tFNNetworkTime, [1188](#)
- tFNNewGPS, [1188](#)
- tFNNewNMEA, [1189](#)
- tFNNewRMTTransferStatistics, [1189](#)
- tFNNewSMS, [1190](#)
- tFNOMADMState, [1190](#)
- tFNOTASPStatus, [1191](#)
- tFNOpMode, [1190](#)
- tFNPDSState, [1191](#)
- tFNPacketSrvState, [1191](#)
- tFNPower, [1191](#)
- tFNPrivacyChange, [1192](#)
- tFNQosNWStatus, [1192](#)
- tFNQosPriEvent, [1192](#)
- tFNQosStatus, [1192](#)
- tFNRInfo, [1194](#)
- tFNRankIndicator, [1193](#)
- tFNResetInfo, [1193](#)
- tFNRoamingIndicator, [1194](#)
- tFNSDKTerminated, [1194](#)
- tFNSLQSOADMAlert, [1195](#)
- tFNSLQSQOSEvent, [1196](#)
- tFNSLQSSessionState, [1196](#)
- tFNSLQSSignalStrengths, [1196](#)
- tFNSLQSWDSEvent, [1196](#)
- tFNSMSEvents, [1197](#)
- tFNSUPSInfo, [1197](#)
- tFNSUPSNotification, [1197](#)
- tFNSensorStreaming, [1194](#)
- tFNServingSystem, [1195](#)
- tFNSetCradleMount, [1195](#)
- tFNSetEngineState, [1195](#)
- tFNSetEventTimeSync, [1195](#)
- tFNSetExtPowerConfig, [1195](#)
- tFNSigInfo, [1195](#)
- tFNSignalStrength, [1195](#)
- tFNSysInfo, [1197](#)
- tFNSysSelectionPref, [1197](#)
- tFNUIMRefresh, [1198](#)
- tFNUIMStatusChangeInfo, [1198](#)
- tFNUSSDNoWaitIndication, [1199](#)
- tFNUSSDNotification, [1198](#)
- tFNUSSDRelease, [1199](#)
- tFNtransLayerInfo, [1198](#)
- tFNtransNWRegInfo, [1198](#)
- THIRD\_INSTANCE, [1165](#)
- transLayerNotification, [1199](#)
- transNWRegInfoNotification, [1199](#)
- USSD\_DCS\_8BIT, [1165](#)
- USSD\_DCS\_ASCII, [1165](#)
- USSD\_DCS\_UCS2, [1165](#)
- VOICE\_SRV, [1165](#)
- WDS\_SRV, [1166](#)
- qaGobiApiDcs.h, [1237](#)
- LEN, [1238](#)
- PORTNAM\_LEN, [1238](#)
- QCWWAN2kConnect, [1238](#)
- QCWWAN2kEnumerateDevices, [1238](#)
- QCWWAN2kGetConnectedDeviceID, [1239](#)
- QCWWANConnect, [1239](#)
- QCWWANDisconnect, [1240](#)
- QCWWANEnumerateDevices, [1240](#)
- SLQSGetDeviceMode, [1241](#)
- SLQSGetNetStatistic, [1241](#)
- SLQSGetUsbPortNames, [1242](#)
- SLQSKillSDKProcess, [1242](#)
- SLQSSetLoggingMask, [1243](#)
- SLQSStart, [1243](#)
- SLQSStart\_AVAgent, [1244](#)
- SLQSStartSrv, [1244](#)
- SetSDKImagePath, [1241](#)
- qaGobiApiDms.h, [1245](#)
- ActivateAutomatic, [1254](#)
- custFeaturesInfo, [1248](#)
- custFeaturesSetting, [1249](#)
- dmsCurrentPRLInfo, [1250](#)
- ERIFileparams, [1251](#)
- GetActivationState, [1255](#)
- GetDeviceCapabilities, [1255](#)
- GetFirmwareRevision, [1256](#)
- GetFirmwareRevisions, [1257](#)
- GetHardwareRevision, [1257](#)
- GetIMSI, [1258](#)
- GetManufacturer, [1258](#)
- GetModelID, [1259](#)
- GetNetworkTime, [1259](#)
- GetOfflineReason, [1260](#)
- GetPRLVersion, [1261](#)
- GetPower, [1261](#)
- GetSerialNumbers, [1261](#)
- GetVoiceNumber, [1262](#)
- IMGDETAILS\_LEN, [1247](#)
- MAX\_CUST\_ID\_LEN, [1247](#)
- MAX\_FSN\_LENGTH, [1247](#)
- ResetToFactoryDefaults, [1263](#)
- SLQSDmsSwiGetResetInfo, [1264](#)
- SLQSDmsSwiIndicationRegister, [1264](#)
- SLQSGetBandCapabilities, [1264](#)
- SLQSGetBandCapability, [1264](#)

- SLQSGetCurrentPRLInfo, 1266
- SLQSGetCustFeatures, 1266
- SLQSGetCustFeaturesV2, 1267
- SLQSGetERIFile, 1267
- SLQSGetSerialNumbers, 1267
- SLQSSetCustFeatures, 1268
- SLQSSetCustFeaturesV2, 1268
- SLQSSwiClearDyingGaspStatistics, 1269
- SLQSSwiGetCrashAction, 1269
- SLQSSwiGetCrashInfo, 1269
- SLQSSwiGetDyingGaspCfg, 1270
- SLQSSwiGetDyingGaspStatistics, 1270
- SLQSSwiGetFSN, 1271
- SLQSSwiGetFirmwareCurr, 1270
- SLQSSwiGetFwUpdateStatus, 1271
- SLQSSwiGetHostDevInfo, 1272
- SLQSSwiGetHostDevInfoParams, 1252
- SLQSSwiGetOSInfo, 1272
- SLQSSwiGetOSInfoParams, 1252
- SLQSSwiGetSerialNoExt, 1272
- SLQSSwiGetSerialNoExtParams, 1253
- SLQSSwiGetUSBComp, 1273
- SLQSSwiSetCrashAction, 1273
- SLQSSwiSetDyingGaspCfg, 1274
- SLQSSwiSetHostDevInfo, 1274
- SLQSSwiSetHostDevInfoParams, 1253
- SLQSSwiSetOSInfo, 1275
- SLQSSwiSetOSInfoParams, 1254
- SLQSSwiSetUSBComp, 1275
- SLQSUIMGetState, 1275
- serialNumbersInfo, 1251
- SetPower, 1263
- UIMChangePIN, 1276
- UIMGetControlKeyStatus, 1277
- UIMGetICCID, 1278
- UIMGetPINStatus, 1278
- UIMSetControlKeyProtection, 1279
- UIMSetPINProtection, 1280
- UIMUnblockControlKey, 1281
- UIMUnblockPIN, 1281
- UIMVerifyPIN, 1282
- UNIQUE\_ID\_LEN, 1247
- ValidateSPC, 1283
- qaGobiApiFms.h, 1283
  - BUILD\_ID\_LEN, 1286
  - DEVICE\_OFFLINE, 1286
  - DEVICE\_RESET, 1286
  - DEVICE\_SHUTDOWN, 1286
  - DeleteStoredImage, 1289
  - eGetDeviceSeries, 1290
  - eGobiDeviceSeries, 1287
  - eGobiImageCarrier, 1287
  - eGobiImageGPS, 1289
  - eGobiImageRegion, 1289
  - eGobiImageTech, 1289
  - GetImageStore, 1291
  - GetImagesPreference, 1290
  - GetStoredImages, 1291
  - IMG\_ID\_LEN, 1287
  - PRI\_UPDATE\_FAIL, 1287
  - SLQSDownloadFirmwareToSlot, 1292
  - SLQSGetBootVersionNumber, 1293
  - SLQSGetFirmwareInfo, 1294
  - SLQSGetImageInfo, 1294
  - SLQSGetImageInfo\_9x15, 1295
  - SLQSGetImageInfoMC77xx, 1295
  - SLQSGetImageInfoMC83xx, 1296
  - SLQSGetValidFwPriCombinations, 1297
  - SLQSLsSpkgFormatRequired, 1297
  - SLQSSetSIMBasedImageSwitching, 1297
  - SLQSSetSpkgFormatRequired, 1298
  - SLQSSwiGetAllCarrierImages, 1298
  - SLQSupgradeFirmware9x15, 1299
  - SetImagesPreference, 1292
  - upgrade\_mc77xx\_fw, 1299
  - UpgradeFirmware2k, 1300
- qaGobiApilms.h, 1300
  - SLQSGetIMSSMSConfig, 1301
  - SLQSGetIMSUserConfig, 1302
  - SLQSGetIMSVoIPConfig, 1302
  - SLQSGetRegMgrConfig, 1303
  - SLQSGetSIPConfig, 1303
  - SLQSLmsConfigIndicationRegister, 1303
  - SLQSSetIMSSMSConfig, 1304
  - SLQSSetIMSUserConfig, 1304
  - SLQSSetIMSVoIPConfig, 1305
  - SLQSSetRegMgrConfig, 1305
  - SLQSSetSIPConfig, 1306
- qaGobiApilmsa.h, 1307
  - SLQSGetIMSAREgStatus, 1307
  - SLQSGetIMSAServiceStatus, 1308
  - SLQSGetIMSASupportedFields, 1308
  - SLQSGetIMSASupportedMsg, 1309
  - SLQSRegisterIMSASIndication, 1309
- qaGobiApiLoc.h, 1310
  - SLQSLOCDeAssData, 1311
  - SLQSLOCEventRegister, 1311
  - SLQSLOCGetBestAvailPos, 1312
  - SLQSLOCInjectPosition, 1312
  - SLQSLOCInjectSensorData, 1312
  - SLQSLOCInjectUTCTime, 1313
  - SLQSLOCSetCradleMountConfig, 1313
  - SLQSLOCSetExtPowerState, 1314
  - SLQSLOCSetOpMode, 1314
  - SLQSLOCStart, 1315
  - SLQSLOCStop, 1315
  - SwiLocGetAutoStart, 1315
  - SwiLocSetAutoStart, 1316
- qaGobiApiNas.h, 1316
  - eSYS\_SRV\_DOMAIN, 1333
  - GetACCOLC, 1334
  - GetANAAAAAuthenticationStatus, 1334
  - GetCDMANetworkParameters, 1335
  - GetHomeNetwork, 1336
  - GetHomeNetwork3GPP2, 1337
  - GetNetworkPreference, 1339

- GetRFInfo, [1340](#)
- GetServingNetwork, [1340](#)
- GetServingNetworkCapabilities, [1341](#)
- GetSignalStrengths, [1342](#)
- IMSI\_M\_S1\_LENGTH, [1321](#)
- IMSI\_M\_S2\_LENGTH, [1321](#)
- InitiateDomainAttach, [1343](#)
- InitiateNetworkRegistration, [1343](#)
- MAX\_PILOT\_SETS, [1321](#)
- NAM\_NAME\_LENGTH, [1321](#)
- PLMN\_LENGTH, [1322](#)
- PerformNetworkScan, [1344](#)
- SLQSConfigSigInfo, [1347](#)
- SLQSGetErrorRate, [1348](#)
- SLQSGetNetworkTime, [1348](#)
- SLQSGetOperatorNameData, [1348](#)
- SLQSGetPLMNName, [1349](#)
- SLQSGetServingSystem, [1349](#)
- SLQSGetSignalStrength, [1350](#)
- SLQSGetSysSelectionPref, [1350](#)
- SLQSInitiateNetworkRegistration, [1351](#)
- SLQSNASGetLTECPHYCaInfo, [1353](#)
- SLQSNASSwiGetChannelLock, [1356](#)
- SLQSNASSwiSetChannelLock, [1358](#)
- SLQSNasConfigSigInfo2, [1351](#)
- SLQSNasGet3GPP2Subscription, [1352](#)
- SLQSNasGetCellLocationInfo, [1352](#)
- SLQSNasGetHDRColorCode, [1353](#)
- SLQSNasGetSigInfo, [1353](#)
- SLQSNasGetSysInfo, [1354](#)
- SLQSNasGetTxRxInfo, [1354](#)
- SLQSNasIndicationRegister, [1355](#)
- SLQSNasIndicationRegisterExt, [1356](#)
- SLQSNasIndicationRegisterLTECphyCa, [1356](#)
- SLQSNasSwiIndicationRegister, [1357](#)
- SLQSNasSwiModemStatus, [1357](#)
- SLQSPerformNetworkScan, [1358](#)
- SLQSSetBandPreference, [1358](#)
- SLQSSetSysSelectionPref, [1360](#)
- SLQSSwiGetHDRPersonality, [1360](#)
- SLQSSwiGetHDRProtSubtype, [1361](#)
- SLQSSwiGetHRPDStats, [1361](#)
- SLQSSwiGetLteCQI, [1361](#)
- SLQSSwiNetworkDebug, [1362](#)
- SLQSSwiPSDetach, [1362](#)
- SetACCOLC, [1344](#)
- SetCDMANetworkParameters, [1345](#)
- SetNetworkPreference, [1346](#)
- SlqsNas3GppNetworkRAT, [1322](#)
- slqsNetworkScanInfo, [1322](#)
- sysSelectPrefInfo, [1323](#)
- sysSelectPrefParams, [1327](#)
- UATISIZE, [1322](#)
- qaGobiApiOmadm.h, [1363](#)
  - OMADMCancelSession, [1363](#)
  - OMADMGetPendingNIA, [1363](#)
  - OMADMGetSessionInfo, [1364](#)
  - OMADMStartSession, [1365](#)
- qaGobiApiPds.h, [1366](#)
  - DEFAULTBYTEVALUE, [1367](#)
  - DEFAULTLONGVALUE, [1367](#)
  - DEFAULTWORDVALUE, [1367](#)
  - ForceXTRADownload, [1367](#)
  - GetPDSDDefaults, [1367](#)
  - GetPDSSState, [1368](#)
  - GetPortAutomaticTracking, [1369](#)
  - GetServiceAutomaticTracking, [1369](#)
  - GetXTRAAutomaticDownload, [1370](#)
  - GetXTRANetwork, [1370](#)
  - GetXTRAValidity, [1371](#)
  - PDSInjectTimeReference, [1371](#)
  - ResetPDSData, [1372](#)
  - SLQSGetAGPSConfig, [1376](#)
  - SLQSGetGPSSStateInfo, [1376](#)
  - SLQSPDSDeterminePosition, [1377](#)
  - SLQSPDSInjectAbsoluteTimeReference, [1377](#)
  - SLQSPDSInjectPositionData, [1378](#)
  - SLQSSetAGPSConfig, [1378](#)
  - SLQSSetPositionMethodState, [1379](#)
  - SetPDSDDefaults, [1372](#)
  - SetPDSSState, [1373](#)
  - SetPortAutomaticTracking, [1374](#)
  - SetServiceAutomaticTracking, [1374](#)
  - SetXTRAAutomaticDownload, [1375](#)
  - SetXTRANetwork, [1375](#)
  - StartPDSTrackingSessionExt, [1379](#)
  - StopPDSTrackingSession, [1380](#)
- qaGobiApiQos.h, [1381](#)
  - SLQSQosGetFlowStatus, [1382](#)
  - SLQSQosGetGranted, [1382](#)
  - SLQSQosGetNWProf, [1383](#)
  - SLQSQosGetNetworkStatus, [1383](#)
  - SLQSQosModify, [1383](#)
  - SLQSQosRel, [1384](#)
  - SLQSQosReq, [1384](#)
  - SLQSQosReset, [1385](#)
  - SLQSQosResume, [1385](#)
  - SLQSQosSuspend, [1386](#)
  - SLQSQosSwiReadApnExtraParams, [1386](#)
  - SLQSQosSwiReadDataStats, [1387](#)
- qaGobiApiRms.h, [1387](#)
  - GetSMSWake, [1387](#)
  - SetSMSWake, [1388](#)
- qaGobiApiSar.h, [1388](#)
  - eQMISARRFState, [1389](#)
  - SLQSGetRfSarState, [1390](#)
  - SLQSSetRfSarState, [1390](#)
- qaGobiApiSms.h, [1391](#)
  - ABSOLUTE\_VALIDITY, [1393](#)
  - CONFIG\_LEN, [1393](#)
  - getIndicationRegResp, [1393](#)
  - GetSMSCAddress, [1396](#)
  - getTransLayerInfoResp, [1394](#)
  - getTransNWRegInfoResp, [1394](#)
  - MAX\_SMS\_ROUTES, [1393](#)
  - NUM\_OF\_SET, [1393](#)

- qaQmi3GPP2BroadcastCfgInfo, 1394
- qaQmi3GPPBroadcastCfgInfo, 1395
- SLQSCDMADecodeMTTextMsg, 1399
- SLQSCDMAEncodeMOTextMsg, 1399
- SLQSDeleteSMS, 1400
- SLQSGetIndicationRegister, 1401
- SLQSGetMessageWaiting, 1401
- SLQSGetSMS, 1402
- SLQSGetSMSList, 1403
- SLQSGetSmsBroadcastConfig, 1403
- SLQSGetTransLayerInfo, 1404
- SLQSGetTransNWRegInfo, 1405
- SLQSModifySMSStatus, 1405
- SLQSSendAsyncSMS, 1406
- SLQSSendLongSMS, 1406
- SLQSSendSMS, 1407
- SLQSSetIndicationRegister, 1408
- SLQSSetSmsBroadcastActivation, 1408
- SLQSSetSmsBroadcastConfig, 1409
- SLQSSetSmsStorage, 1409
- SLQSSmsGetMaxStorageSize, 1410
- SLQSSmsGetMessageProtocol, 1410
- SLQSSmsSetRoutes, 1411
- SLQSSwiGetSMSStorage, 1411
- SLQSWCDMADecodeLongTextMsg, 1412
- SLQSWCDMADecodeMTTextMsg, 1412
- SLQSWCDMAEncodeMOTextMsg, 1412
- SaveSMS, 1397
- SendSMS, 1398
- setIndicationRegReq, 1395
- SetSMSCAddress, 1398
- TIME\_DATE\_BUF, 1393
- TIME\_STAMP\_BUF, 1393
- transLayerInfo, 1396
- qaGobiApiSwi.h, 1413
  - SLQSGetPidof, 1413
  - SLQSGetSdkVersion, 1414
  - SLQSSendRawQMI, 1414
- qaGobiApiSwiAudio.h, 1414
  - SLQSGetM2MAVMMute, 1416
  - SLQSGetM2MAudioProfile, 1415
  - SLQSGetM2MAudioVolume, 1416
  - SLQSGetM2MSpkrGain, 1416
  - SLQSSetM2MAVMMute, 1419
  - SLQSSetM2MAudioAVCFG, 1417
  - SLQSSetM2MAudioLPBK, 1417
  - SLQSSetM2MAudioNVDef, 1418
  - SLQSSetM2MAudioProfile, 1418
  - SLQSSetM2MAudioVolume, 1418
  - SLQSSetM2MSpkrGain, 1419
- qaGobiApiSwiOmadms.h, 1420
  - SLQSOMADMCancelSession, 1425
  - SLQSOMADMGetSessionInfo, 1425
  - SLQSOMADMGetSettings, 1426
  - SLQSOMADMGetSettings2, 1426
  - SLQSOMADMGetSettings3, 1424
  - SLQSOMADMStartSession, 1429
  - SLQSOMADMStartSession2, 1430
- SLQSOMADMSetSettings, 1428
- SLQSOMADMSetSettings2, 1429
- SLQSOMADMSetSettings3, 1429
- SLQSOMADMSettings, 1422
- SLQSOMADMSettingsReqParams, 1423
- SLQSOMADMSettingsReqParams3, 1424
- SLQSOMADMStartSession, 1429
- SLQSOMADMStartSession2, 1430
- qaGobiApiTableBandClasses.h, 1430
- qaGobiApiTableCallControlReturnReasons.h, 1433
- qaGobiApiTableCallEndReasons.h, 1434
- qaGobiApiTableCarrierCodes.h, 1449
- qaGobiApiTableCodingScheme.h, 1451
- qaGobiApiTableGpsCapabilityCodes.h, 1454
- qaGobiApiTablePowerModes.h, 1454
- qaGobiApiTableRadioInterfaces.h, 1455
- qaGobiApiTableRegionCodes.h, 1456
- qaGobiApiTableServiceOptions.h, 1456
- qaGobiApiTableSupServiceInfoClasses.h, 1458
- qaGobiApiTableSwiAudio.h, 1459
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, 1459
- qaGobiApiTableVoiceCallEndReasons.h, 1461
- qaGobiApiTmd.h, 1467
  - SLQSTmdDeRegNotMitigationLvl, 1468
  - SLQSTmdGetMitigationDevList, 1468
  - SLQSTmdGetMitigationLvl, 1469
  - SLQSTmdRegNotMitigationLvl, 1469
- qaGobiApiUim.h, 1469
  - MAX\_ICCID\_LENGTH, 1472
  - MAX\_NO\_OF\_SLOTS, 1472
  - MAX\_PATH\_LENGTH, 1472
  - MAX\_PUK\_LENGTH, 1472
  - MAX\_SLOTS\_STATUS, 1472
  - SLQSUIMAuthenticate, 1472
  - SLQSUIMChangePin, 1472
  - SLQSUIMDepersonalization, 1473
  - SLQSUIMEventRegister, 1473
  - SLQSUIMGetCardStatus, 1474
  - SLQSUIMGetConfiguration, 1474
  - SLQSUIMGetFileAttributes, 1475
  - SLQSUIMGetSlotsStatus, 1476
  - SLQSUIMPowerDown, 1476
  - SLQSUIMPowerUp, 1476
  - SLQSUIMReadTransparent, 1477
  - SLQSUIMRefreshComplete, 1477
  - SLQSUIMRefreshGetLastEvent, 1478
  - SLQSUIMRefreshOK, 1478
  - SLQSUIMRefreshRegister, 1479
  - SLQSUIMReset, 1479
  - SLQSUIMSetPinProtection, 1480
  - SLQSUIMSwitchSlot, 1480
  - SLQSUIMUnblockPin, 1481
  - SLQSUIMVerifyPin, 1481
- qaGobiApiVoice.h, 1482
  - AnswerUSSD, 1486
  - CancelUSSD, 1486
  - MAX\_CALL\_NO\_LEN, 1485

- MAX\_NO\_OF\_CALLS, 1485
- MAXUSSDLENGTH, 1485
- OriginateUSSD, 1486
- PASSWORD\_LENGTH, 1485
- SLQSOriginateUSSD, 1487
- SLQSVoiceALSSelectLine, 1487
- SLQSVoiceALSSetLineSwitching, 1488
- SLQSVoiceAnswerCall, 1488
- SLQSVoiceBindSubscription, 1489
- SLQSVoiceBurstDTMF, 1489
- SLQSVoiceDialCall, 1490
- SLQSVoiceEndCall, 1490
- SLQSVoiceGetAllCallInfo, 1491
- SLQSVoiceGetCLIP, 1493
- SLQSVoiceGetCLIR, 1494
- SLQSVoiceGetCNAP, 1494
- SLQSVoiceGetCOLP, 1495
- SLQSVoiceGetCOLR, 1495
- SLQSVoiceGetCallBarring, 1491
- SLQSVoiceGetCallForwardingStatus, 1492
- SLQSVoiceGetCallInfo, 1492
- SLQSVoiceGetCallWaiting, 1493
- SLQSVoiceGetConfig, 1496
- SLQSVoiceIndicationRegister, 1497
- SLQSVoiceManageCalls, 1497
- SLQSVoiceOrigUSSDNoWait, 1498
- SLQSVoiceSendFlash, 1498
- SLQSVoiceSetCallBarringPassword, 1499
- SLQSVoiceSetConfig, 1499
- SLQSVoiceSetPreferredPrivacy, 1500
- SLQSVoiceSetSUPSService, 1500
- SLQSVoiceStartContDTMF, 1501
- SLQSVoiceStopContDTMF, 1501
- serviceClassInformation, 1485
- qaGobiApiWds.h, 1502
  - GetAutoconnect, 1509
  - GetByteTotals, 1509
  - GetConnectionRate, 1510
  - GetDataBearerTechnology, 1510
  - GetDefaultProfile, 1511
  - GetDefaultProfileLTE, 1513
  - GetDefaultProfileNum, 1514
  - GetDormancyState, 1515
  - GetIPAddressLTE, 1515
  - GetLastMobileIPError, 1516
  - GetMobileIP, 1516
  - GetMobileIPProfile, 1517
  - GetPacketStatistics, 1518
  - GetPacketStatus, 1519
  - GetProfileSettingIn, 1506
  - GetProfileSettingOut, 1506
  - GetSessionDuration, 1519
  - GetSessionState, 1520
  - iGetByteTotals, 1520
  - iGetConnectionRate, 1521
  - iGetPacketStatistics, 1521
  - iSLQSMISetIPFamilyPreference, 1521
  - qmiDataBearerMasks, 1509
  - QmiProfileInfo, 1506
  - QmiWDSDataBearerTechnology, 1507
  - QmiWDSDataBearers, 1506
  - RMSetTransferStatistics, 1521
  - SLQSAutoConnect, 1529
  - SLQSCreateProfile, 1530
  - SLQSDeleteProfile, 1530
  - SLQSGet3GPPConfigItem, 1531
  - SLQSGetByteTotals, 1531
  - SLQSGetConnectionRate, 1532
  - SLQSGetCurrDataSystemStat, 1532
  - SLQSGetCurrentChannelRate, 1533
  - SLQSGetDUNCallInfo, 1534
  - SLQSGetDataBearerTechnology, 1533
  - SLQSGetDataBearerTechnologyExt, 1534
  - SLQSGetPacketStatistics, 1534
  - SLQSGetProfile, 1535
  - SLQSGetProfileSettings, 1536
  - SLQSGetRuntimeSettings, 1537
  - SLQSGetSessionState, 1537
  - SLQSModifyProfile, 1538
  - SLQSResetPacketStatics, 1539
  - SLQSSetDHCPv4ClientConfig, 1541
  - SLQSSetLoopback, 1541
  - SLQSSetDHCPv4ClientConfig, 1541
  - SLQSSetLoopback, 1542
  - SLQSSet3GPPConfigItem, 1539
  - SLQSSetProfile, 1539
  - SLQSSetStopDataSession, 1542
  - SLQSWdsGoActive, 1542
  - SLQSWdsGoDormant, 1543
  - SLQSWdsSetEventReport, 1543
  - SLQSWdsSwiPDPRuntimeSettings, 1544
  - SetActiveMobileIPProfile, 1521
  - SetAutoconnect, 1521
  - SetDefaultProfile, 1522
  - SetDefaultProfileLTE, 1523
  - SetDefaultProfileLTEV2, 1525
  - SetDefaultProfileNum, 1526
  - SetMobileIP, 1527
  - SetMobileIPParameters, 1527
  - SetMobileIPProfile, 1528
  - slqs3GPPConfigItem, 1507
  - WDS\_IsGobiDevice, 1544
- qaNasGetRFBandInfo.h, 1544
  - PkQmiNasGetRFBandInfo, 1545
  - UpkQmiNasGetRFBandInfo, 1545
- qaNasPerformNetworkScan.h, 1545
  - FORBIDDEN\_INDEX, 1546
  - INDEX\_ZERO, 1546
  - PREFERRED\_INDEX, 1546
  - PkQmiNasPerformNetworkScan, 1546
  - ROAMING\_INDEX, 1546
  - UpkQmiNasPerformNetworkScan, 1546
- qaQmi3GPP2BroadcastCfgInfo
  - qaGobiApiSms.h, 1394
- qaQmi3GPPBroadcastCfgInfo
  - qaGobiApiSms.h, 1395

- qaQmi3Gpp2TimeZone, [614](#)
  - daylightSavings, [615](#)
  - leapSeconds, [615](#)
  - localTimeOffset, [615](#)
- qaQmiInterfaceInfo, [615](#)
  - qaQmiinstanceid, [615](#)
  - qaQmisvctype, [615](#)
  - v4sessionId, [615](#)
  - v6sessionId, [615](#)
- qaQmiServingSystemParam, [615](#)
  - BasestationID, [618](#)
  - BasestationLatitude, [618](#)
  - BasestationLongitude, [618](#)
  - CDMA\_P\_Rev, [619](#)
  - CDMASystemInfoExt, [619](#)
  - CallBarStatus, [618](#)
  - CellID, [619](#)
  - concSvcInfo, [619](#)
  - CurrentPLMN, [619](#)
  - DTMInd, [619](#)
  - DataSrvCapabilities, [619](#)
  - defaultRoamInd, [619](#)
  - DetailedSvcInfo, [619](#)
  - Gpp2TimeZone, [619](#)
  - GppNetworkDSTAdjustment, [619](#)
  - GppTimeZone, [619](#)
  - hdrPersonality, [619](#)
  - Lac, [619](#)
  - NetworkID, [619](#)
  - PRLInd, [619](#)
  - roamIndicatorVal, [619](#)
  - RoamingIndicatorList, [619](#)
  - ServingSystem, [619](#)
  - SystemID, [619](#)
  - trackAreaCode, [619](#)
- qaQmiinstanceid
  - qaQmiInterfaceInfo, [615](#)
- qaQmisvctype
  - qaQmiInterfaceInfo, [615](#)
- qm\_wds\_ds\_profile\_extended\_err\_codes
  - qmerrno.h, [1552](#)
- qmerrno.h
  - eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT, [1549](#)
  - eQCWWAN\_ERR\_BUFFER\_SZ, [1548](#)
  - eQCWWAN\_ERR\_CANCEL\_OP, [1549](#)
  - eQCWWAN\_ERR\_DRIVER, [1549](#)
  - eQCWWAN\_ERR\_ENUM\_BEGIN, [1548](#)
  - eQCWWAN\_ERR\_ENUM\_END, [1549](#)
  - eQCWWAN\_ERR\_FILE\_COPY, [1548](#)
  - eQCWWAN\_ERR\_FILE\_OPEN, [1548](#)
  - eQCWWAN\_ERR\_GENERAL, [1548](#)
  - eQCWWAN\_ERR\_INTERNAL, [1548](#)
  - eQCWWAN\_ERR\_INVALID\_ARG, [1548](#)
  - eQCWWAN\_ERR\_INVALID\_DEVID, [1548](#)
  - eQCWWAN\_ERR\_INVALID\_FILE, [1548](#)
  - eQCWWAN\_ERR\_INVALID\_QMI\_RSP, [1548](#)
  - eQCWWAN\_ERR\_INVALID\_XID, [1549](#)
  - eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP, [1548](#)
  - eQCWWAN\_ERR\_MEMORY, [1548](#)
  - eQCWWAN\_ERR\_MULTIPLE\_DEVICES, [1549](#)
  - eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPOR-  
TED, [1549](#)
  - eQCWWAN\_ERR\_NO\_CANCELABLE\_OP, [1549](#)
  - eQCWWAN\_ERR\_NO\_CONNECTION, [1548](#)
  - eQCWWAN\_ERR\_NO\_DEVICE, [1548](#)
  - eQCWWAN\_ERR\_NO\_SIGNAL, [1549](#)
  - eQCWWAN\_ERR\_NONE, [1548](#)
  - eQCWWAN\_ERR\_NULL\_TLV, [1552](#)
  - eQCWWAN\_ERR\_OFFLINE, [1549](#)
  - eQCWWAN\_ERR\_PDU\_GENERATION, [1549](#)
  - eQCWWAN\_ERR\_QMI\_ABORTED, [1549](#)
  - eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED, [1551](#)
  - eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT, [1551](#)
  - eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG, [1549](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAI-  
LED, [1550](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LO-  
CK, [1550](#)
  - eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPP-  
ORTED, [1551](#)
  - eQCWWAN\_ERR\_QMI\_CALL\_FAILED, [1549](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP, [1552](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTRO-  
L\_FAILED, [1551](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_END, [1552](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_START, [1552](#)
  - eQCWWAN\_ERR\_QMI\_CAUSE\_CODE, [1550](#)
  - eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUST-  
ED, [1549](#)
  - eQCWWAN\_ERR\_QMI\_CONNECT, [1548](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE, [1549](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY, [1550](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FU-  
LL, [1550](#)
  - eQCWWAN\_ERR\_QMI\_DISABLED, [1550](#)
  - eQCWWAN\_ERR\_QMI\_ENCODING, [1550](#)
  - eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAI-  
LURE, [1552](#)
  - eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED, [1552](#)
  - eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL, [1551](#)
  - eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT, [1551](#)
  - eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED, [1550](#)
  - eQCWWAN\_ERR\_QMI\_GENERAL, [1550](#)
  - eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRIC-  
TED, [1551](#)
  - eQCWWAN\_ERR\_QMI\_IFACE, [1548](#)
  - eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE, [1551](#)
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FI-  
LTER, [1550](#)

- eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN, [1549](#)
- eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE, [1551](#)
- eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT, [1551](#)
- eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES, [1550](#)
- eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND, [1550](#)
- eQCWWAN\_ERR\_QMI\_INTERNAL, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ARG, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD, [1552](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ID, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_INDEX, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION, [1551](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTION, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD, [1551](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF, [1549](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP, [1552](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION, [1550](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID, [1549](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG, [1549](#)
- eQCWWAN\_ERR\_QMI\_MAX, [1551](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE, [1550](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_USE, [1550](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAILURE, [1550](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT, [1550](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG, [1549](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED, [1551](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED, [1551](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY, [1550](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE, [1550](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY, [1550](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND, [1549](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO, [1551](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION, [1551](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS, [1549](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE, [1550](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED, [1549](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED, [1551](#)
- eQCWWAN\_ERR\_QMI\_OFFSET, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED, [1549](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE, [1551](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL, [1549](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED, [1550](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED, [1550](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH, [1551](#)
- eQCWWAN\_ERR\_QMI\_REQ, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO, [1548](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUPPORTED, [1550](#)
- eQCWWAN\_ERR\_QMI\_RSP, [1548](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO, [1548](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER, [1551](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG, [1551](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE, [1550](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID, [1550](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP, [1550](#)

- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND, [1551](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED, [1550](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR, [1551](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE, [1551](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE, [1551](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION, [1549](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN, [1550](#)
- eQCWWAN\_ERR\_QMI\_WIDTH, [1552](#)
- eQCWWAN\_ERR\_RESET, [1549](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR, [1551](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS, [1551](#)
- eQCWWAN\_ERR\_SWICM\_END, [1552](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID, [1551](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID, [1551](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED, [1551](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED, [1551](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED, [1551](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS, [1551](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE, [1551](#)
- eQCWWAN\_ERR\_SWICM\_START, [1551](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN, [1551](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP, [1551](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_END, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR, [1552](#)
- eQCWWAN\_ERR\_SWIDCS\_START, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_END, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOAD\_MODE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT\_ACTIVE\_IMAGE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE, [1552](#)
- eQCWWAN\_ERR\_SWIIM\_START, [1552](#)
- eQCWWAN\_ERR\_SWISM\_END, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED, [1552](#)
- eQCWWAN\_ERR\_SWISMS\_START, [1552](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_END, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM, [1553](#)

- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LEN\_INVALID, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NOT\_INITED, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL, [1553](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END, [1553](#)
- qmerrno.h, [1546](#)
  - eQCWWANError, [1548](#)
  - qm\_wds\_ds\_profile\_extended\_err\_codes, [1552](#)
- QmiCbkCatEventStatusReportInd, [619](#)
  - CCETlv, [619](#)
  - event\_Index, [619](#)
- QmiCbkLocBestAvailPosInd, [620](#)
  - pAltitudeWrtEllipsoid, [624](#)
  - pAltitudeWrtMeanSeaLevel, [624](#)
  - pGpsTime, [624](#)
  - pHeading, [624](#)
  - pHeadingUnc, [624](#)
  - pHorCirConf, [624](#)
  - pHorEllpConf, [624](#)
  - pHorReliability, [624](#)
  - pHorUncCircular, [624](#)
  - pHorUncEllipseOrientAzimuth, [624](#)
  - pHorUncEllipseSemiMajor, [624](#)
  - pHorUncEllipseSemiMinor, [624](#)
  - pLatitude, [625](#)
  - pLongitude, [625](#)
  - pMagneticDeviation, [625](#)
  - pPrecisionDilution, [625](#)
  - pSensorDataUsage, [625](#)
  - pSpeedHorizontal, [625](#)
  - pSpeedUnc, [625](#)
  - pSpeedVertical, [625](#)
  - pSpeedVerticalUnc, [625](#)
  - pSvUsedforFix, [625](#)
  - pTechnologyMask, [625](#)
  - pTimeSrc, [625](#)
  - pTimeUnc, [625](#)
  - pTimestampUtc, [625](#)
  - pVertConfidence, [625](#)
  - pVertReliability, [625](#)
  - pVertUnc, [625](#)
  - pXid, [625](#)
  - status, [625](#)
- QmiCbkLocCradleMountInd, [625](#)
  - cradleMountConfigStatus, [626](#)
- QmiCbkLocEngineStateInd, [626](#)
  - engineState, [626](#)
- QmiCbkLocEventTimeSyncInd, [626](#)
  - timeSyncRefCounter, [627](#)
- QmiCbkLocInjectPositionInd, [627](#)
  - status, [627](#)
- QmiCbkLocInjectSensorDataInd, [628](#)
  - injectSensorDataStatus, [629](#)
  - pAccelSamplesAccepted, [629](#)
  - pAccelTempSamplesAccepted, [629](#)
  - pGyroSamplesAccepted, [629](#)
  - pGyroTempSamplesAccepted, [629](#)
  - pOpaqueIdentifier, [629](#)
- QmiCbkLocInjectTimeInd, [629](#)
  - injectTimeSyncStatus, [629](#)
- QmiCbkLocInjectUTCTimeInd, [629](#)
  - status, [630](#)
- QmiCbkLocPositionReportInd, [630](#)
  - pAltitudeAssumed, [635](#)
  - pAltitudeWrtEllipsoid, [635](#)
  - pAltitudeWrtMeanSeaLevel, [635](#)
  - pFixId, [635](#)
  - pGpsTime, [635](#)
  - pHeading, [635](#)
  - pHeadingUnc, [635](#)
  - pHorConfidence, [635](#)
  - pHorReliability, [635](#)
  - pHorUncCircular, [635](#)
  - pHorUncEllipseOrientAzimuth, [635](#)
  - pHorUncEllipseSemiMajor, [635](#)
  - pHorUncEllipseSemiMinor, [635](#)
  - pLatitude, [635](#)
  - pLeapSeconds, [635](#)
  - pLongitude, [635](#)
  - pMagneticDeviation, [635](#)
  - pPrecisionDilution, [635](#)
  - pSensorDataUsage, [635](#)
  - pSpeedHorizontal, [635](#)
  - pSpeedUnc, [635](#)
  - pSpeedVertical, [635](#)
  - pSvUsedforFix, [635](#)
  - pTechnologyMask, [635](#)
  - pTimeSrc, [635](#)
  - pTimeUnc, [635](#)
  - pTimestampUtc, [635](#)
  - pVertConfidence, [635](#)
  - pVertReliability, [636](#)
  - pVertUnc, [636](#)
  - sessionId, [636](#)
  - sessionStatus, [636](#)
- QmiCbkLocSensorStreamingInd, [636](#)
  - pAccelAcceptReady, [636](#)
  - pAccelTempAcceptReady, [636](#)
  - pGyroAcceptReady, [636](#)
  - pGyroTempAcceptReady, [636](#)
- QmiCbkLocSetExtPowerConfigInd, [636](#)
  - status, [637](#)
- QmiCbkNasLTECphyCaInfo, [637](#)
  - sPhyCaAggPcellInfo, [638](#)
  - sPhyCaAggScellDIBw, [638](#)
  - sPhyCaAggScellIndType, [638](#)
  - sPhyCaAggScellIndex, [638](#)
  - sPhyCaAggScellInfo, [638](#)
- QmiCbkSwiOmaDmEventStatusReportInd, [638](#)

- SITlv, 638
- QmiCbkSwiOmaDmEventStatusReportIndExt, 638
  - SITlv, 638
- QmiCbkTmdMitiLvIRptInd, 638
  - currentMitigationLvl, 639
  - MitigationDevInfo, 639
- QmiCbkWdsStatisticsIndState, 639
  - RxDropConutTlv, 639
  - RxOkByteCountTlv, 640
  - RxOkConutTlv, 640
  - TxDropConutTlv, 640
  - TxOkByteCountTlv, 640
  - TxOkConutTlv, 640
- qmiDataBearerMasks
  - qaGobiApiWds.h, 1509
- QmiNas3GppNetworkInfo, 641
  - pDescription, 642
  - pForbidden, 642
  - pInUse, 642
  - pMCC, 642
  - pMNC, 642
  - pPreferred, 642
  - pRoaming, 642
- QmiNasGetRFBandInfoResp, 642
  - pInstancesSize, 642
  - pRFBandInfoElements, 642
  - results, 642
- QmiNasPerformNetworkScanResp, 642
  - pInstanceSize, 643
  - pInstances, 643
  - results, 643
- QmiProfileInfo
  - qaGobiApiWds.h, 1506
- qmiSmsMessageList, 643
  - messageIndex, 643
  - messageTag, 643
- QmiWSDDataBearerTechnology
  - qaGobiApiWds.h, 1507
- qmiWSDDataBearerTechnology, 643
  - currentNetwork, 643
  - ratMask, 643
  - soMask, 644
- QmiWSDDataBearers
  - qaGobiApiWds.h, 1506
- QmiWdsIpAddressInfo, 644
  - piPAAddressV4, 644
  - piPAAddressV6, 644
  - piPv6prefixlen, 644
- qmiWdsRunTimeSettings, 644
  - pAPNName, 647
  - pAuthentication, 647
  - pDomainList, 647
  - pGPRSGrantedQoS, 647
  - pGWAddressV4, 647
  - pIMCNflag, 647
  - piPAAddressV4, 647
  - piPFamilyPreference, 647
  - piPV6AddrInfo, 647
  - piPV6GWAddrInfo, 647
  - pMtu, 647
  - pPCSCFAddrPCO, 647
  - pPCSCFFQDNAddrList, 647
  - pPDPTType, 647
  - pPrimaryDNSV4, 648
  - pPrimaryDNSV6, 648
  - pProfileID, 648
  - pProfileName, 648
  - pSecondaryDNSV4, 648
  - pSecondaryDNSV6, 648
  - pServerAddrList, 648
  - pSubnetMaskV4, 648
  - pTechnology, 648
  - pUMTSGrantedQoS, 648
  - pUsername, 648
- qmifwinfo\_s, 640
  - dev, 640
  - g, 641
  - s, 641
- qos.h, 1553
  - pack\_qos\_SLQSQoSGetNetworkStatus, 1555
  - pack\_qos\_SLQSQoSSwiReadApnExtraParams, 1555
  - pack\_qos\_SLQSQoSSwiReadDataStats, 1556
  - pack\_qos\_SLQSSetQoSEventCallback, 1557
  - unpack\_qos\_SLQSQoSGetNetworkStatus, 1557
  - unpack\_qos\_SLQSQoSSwiReadApnExtraParams, 1558
  - unpack\_qos\_SLQSQoSSwiReadDataStats, 1558
  - unpack\_qos\_SLQSSetQoSEventCallback, 1559
  - unpack\_qos\_SLQSSetQoSEventCallback\_ind, 1559
  - unpack\_qos\_SLQSSetQoSNWStatusCallback\_ind, 1559
  - unpack\_qos\_SLQSSetQoSPrEventCallback\_ind, 1560
  - unpack\_qos\_SLQSSetQoSStatusCallback\_ind, 1561
- qos\_id
  - QosMap, 653
- QosClassID, 648
  - gDIBitRate, 649
  - gUIBitRate, 649
  - maxDIBitRate, 649
  - maxUIBitRate, 649
  - QCI, 649
- qosDeliveryOrder
  - LibPackUMTSQoS, 334
  - UMTSMinQoS, 838
  - UMTSQoS, 840
  - wds\_UMTSMinQoS, 1050
- QosEventInfo, 649
  - pDataBearer, 650
  - pPacketsCountRX, 650
  - pPacketsCountTX, 650
  - pTotalBytesRX, 650
  - pTotalBytesTX, 650

- qosFlow
  - sQosStat, [748](#)
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [926](#)
- QosFlowInfo, [651](#)
  - pBearerID, [651](#)
  - pQFlowState, [651](#)
  - pRxQFilter, [651](#)
  - pRxQFlowGranted, [651](#)
  - pTxQFilter, [651](#)
  - pTxQFlowGranted, [651](#)
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, [926](#)
- QosFlowInfoState, [652](#)
  - id, [652](#)
  - isNewFlow, [652](#)
  - state, [652](#)
- QosMap, [652](#)
  - dscp, [653](#)
  - qos\_id, [653](#)
  - state, [653](#)
- Quality of Service (QOS), [48](#)
- RAN
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- RAT
  - \_SlqsNas3GppNetworkRAT\_, [67](#)
  - nas\_QmiNas3GppNetworkRAT, [441](#)
- RATMask
  - CurrNetworkInfo, [178](#)
  - currNetworkInfo, [179](#)
  - wds\_currNetworkInfo, [1043](#)
- REGISTER\_EVENT
  - qaGobiApiCbK.h, [1165](#)
- REGISTER\_SRV
  - qaGobiApiCbK.h, [1165](#)
- RFBandInfoElements, [660](#)
  - activeBandClass, [661](#)
  - activeChannel, [661](#)
  - radioInterface, [661](#)
  - unpack\_nas\_GetRFInfo\_t, [887](#)
- RFTlv
  - unpack\_nas\_SetEventReportInd\_t, [891](#)
- RMAutoConnect
  - pack\_dms\_SetCustFeature\_t, [522](#)
  - unpack\_dms\_GetCustFeature\_t, [845](#)
- RMSetTransferStatistics
  - qaGobiApiWds.h, [1521](#)
- ROAMING\_INDEX
  - qaNasPerformNetworkScan.h, [1546](#)
- RPCause
  - SMSAsyncRawSend\_s, [732](#)
- RPTlv
  - NASQmiCbK NasSystemSelPrefInd, [493](#)
- RRTlv
  - unpack\_nas\_SetEventReportInd\_t, [891](#)
- RSRPThresListLen
  - RSRPThresh, [664](#)
- RSRPThresh, [663](#)
  - pRSRPThresList, [664](#)
- RSRPThresListLen, [664](#)
- RSRQThresListLen
  - RSRQThresh, [665](#)
- RSRQThresh, [664](#)
  - pRSRQThresList, [665](#)
  - RSRQThresListLen, [665](#)
- RSSIThresListLen
  - RSSIThresh, [666](#)
- RSSIThresh, [665](#)
  - pRSSIThresList, [666](#)
  - RSSIThresListLen, [666](#)
- RX\_EC\_IO
  - NetworkStat1x, [506](#)
- RX\_PWR
  - NetworkStat1x, [506](#)
  - NetworkStatEVDO, [508](#)
- RXAGCList, [666](#)
  - pRXAIG, [667](#)
  - pRXComprSlope, [667](#)
  - pRXComprThres, [667](#)
  - pRXExpSlope, [667](#)
  - pRXExpThres, [667](#)
  - pRXStaticGain, [667](#)
- RXAVCList, [667](#)
  - pAVRXAVCHheadroom, [667](#)
  - pAVRXAVCSens, [667](#)
- RXChan
  - LTEInfo, [363](#)
  - nas\_LTEInfo, [419](#)
- rXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- RXOKBytesCount
  - DUNCallInfoInd, [209](#)
- rXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- rXOkBytesCount
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- RXPCMIIRFtr, [669](#)
  - pFlag, [670](#)
  - pStage0Val, [670](#)
  - pStage1Val, [670](#)
  - pStage2Val, [670](#)
  - pStage3Val, [670](#)
  - pStage4Val, [670](#)
  - pStageCnt, [670](#)
- rXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- rXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- rXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- radio
  - unpack\_nas\_GetSignalStrengths\_t, [889](#)
- radio\_if
  - nasGetTxRxInfoReq, [476](#)
- radioIf
  - ecioListElement, [211](#)
  - errorRateListElement, [214](#)

- nas\_ecioListElement, [401](#)
- nas\_errorRateListElement, [402](#)
- nas\_rsrqInformation, [444](#)
- nas\_rxSignalStrengthListElement, [445](#)
- rsrqInformation, [664](#)
- rxSignalStrengthListElement, [671](#)
- Radiolfaces
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
  - unpack\_dms\_GetDeviceCapabilities\_t, [847](#)
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- RadiolfacesSize
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- radiolfacesSize
  - unpack\_dms\_GetDeviceCapabilities\_t, [847](#)
- radiolInterface
  - nas\_RFInfoTlv, [443](#)
  - nas\_roamIndList, [443](#)
  - nas\_servSystem, [446](#)
  - nas\_SignalStrengthTlv, [447](#)
  - nas\_timeInfo, [456](#)
  - RFBandInfoElements, [661](#)
  - roamIndList, [662](#)
  - servSystem, [681](#)
  - timeInfo, [784](#)
- radiolInterfaceList
  - NASServingSystemInfo, [494](#)
  - ServingSystemInfo, [680](#)
- radiolInterfaceNo
  - NASServingSystemInfo, [495](#)
  - ServingSystemInfo, [680](#)
- radiolInterfaceSize
  - nas\_RFInfoTlv, [443](#)
- range
  - Port, [597](#)
  - unpack\_qos\_Port\_t, [921](#)
- RankIndicatorInd, [653](#)
  - Count1, [653](#)
  - Count2, [653](#)
- rat
  - CSGID, [170](#)
  - MNRInfo, [382](#)
  - nas\_CSGID, [398](#)
  - nas\_MNRInfo, [432](#)
- ratMask
  - dataBearerTechnology, [189](#)
  - qmiWSDDataBearerTechnology, [643](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- ratValue
  - DataBearerTech, [187](#)
- rawLen
  - fileAttributes, [219](#)
- rawValue
  - fileAttributes, [219](#)
- rcv4
  - ssdatasession\_params, [752](#)
- rcv6
  - ssdatasession\_params, [752](#)
- readResult, [653](#)
  - content, [654](#)
  - contentLen, [654](#)
- readTransparent
  - pack\_uim\_ReadTransparent\_t, [561](#)
  - UIMReadTransparentReq, [820](#)
- readTransparentInfo, [654](#)
  - length, [654](#)
  - offset, [654](#)
- Reason
  - voiceGetCallFWReq, [996](#)
  - voiceSetCallBarringPwdInfo, [1018](#)
- reason
  - ccSUPSType, [140](#)
  - redirNumInfo, [656](#)
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [928](#)
  - voiceGetCallBarringReq, [994](#)
  - voiceSetSUPSServiceReq, [1025](#)
- receiptAction
  - smsRouteEntry, [744](#)
- receivedBytes
  - omaDmFotaTlvExt, [518](#)
- reconfigReqd
  - \_packetSrvStatus, [62](#)
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [969](#)
- reconfiguration\_required
  - slqsSessionStateInfo, [723](#)
- recordCount
  - fileAttributes, [219](#)
- recordSize
  - fileAttributes, [219](#)
- redirNumInfo, [654](#)
  - numLen, [656](#)
  - numPlan, [656](#)
  - numType, [656](#)
  - number, [656](#)
  - PI, [656](#)
  - reason, [656](#)
  - SI, [656](#)
- RedirPartyNum
  - arrRedirPartyNum, [112](#)
- refData
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [865](#)
- refString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [865](#)
- ReferenceID
  - CatAlPhalIdentifierTlv, [137](#)
  - CatEventIDDDataTlv, [138](#)
- refpn
  - CDMAInfo, [143](#)
  - nas\_CDMAInfo, [389](#)
- refreshComplete
  - UIMRefreshCompleteReq, [822](#)
- RefreshMode
  - CatRefreshTlv, [139](#)

- RefreshStage
  - CatRefreshTlv, [139](#)
- regAction
  - nasInitNetworkReg, [481](#)
  - pack\_nas\_SLQSIInitiateNetworkRegistration\_t, [535](#)
- RegForeignNID
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- RegForeignSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- RegHomeSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- regInd
  - \_transLayerInfoNotification, [92](#)
- regPrd
  - AddCDMASysInfo, [96](#)
  - nas\_AddCDMASysInfo, [385](#)
- regRefresh
  - UIMRefreshRegisterReq, [825](#)
- regRejectInfoValid
  - GSMSysInfo, [272](#)
  - LTESysInfo, [377](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [430](#)
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1041](#)
- regState
  - nas\_servSystem, [446](#)
  - servSystem, [681](#)
- Region
  - fwinfo\_s, [226](#)
- registerFlag
  - registerRefresh, [657](#)
- registerRefresh, [656](#)
  - arrfileInfo, [657](#)
  - numFiles, [657](#)
  - registerFlag, [657](#)
  - voteForInit, [657](#)
- RegistrationState
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- registrationState
  - NASServingSystemInfo, [495](#)
  - ServingSystemInfo, [680](#)
- rejCause
  - GSMSysInfo, [272](#)
  - LTESysInfo, [377](#)
  - nas\_GSMSysInfo, [410](#)
  - nas\_LTESysInfo, [430](#)
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1041](#)
- rejectCause
  - nas\_RejectReasonTlv, [442](#)
- rejectSrvDomain
  - GSMSysInfo, [272](#)
  - LTESysInfo, [377](#)
  - nas\_GSMSysInfo, [411](#)
  - nas\_LTESysInfo, [430](#)
  - nas\_WCDMASysInfo, [467](#)
  - WCDMASysInfo, [1041](#)
- reliabilityClass
  - GPRSQoS, [261](#)
  - GPRSRequestedQoS, [262](#)
  - LibPackGPRSRequestedQoS, [307](#)
  - wds\_GPRSQoS, [1045](#)
- remPartyNumber
  - remotePartyNum, [659](#)
- remainingRetries, [657](#)
  - unblockLeft, [657](#)
  - verifyLeft, [657](#)
- Remote Management Service (RMS), [39](#)
- RemotePartyName
  - getAllCallRmtPtyName, [230](#)
- remotePartyName, [657](#)
  - callerName, [658](#)
  - codingScheme, [658](#)
  - nameLen, [658](#)
  - namePI, [658](#)
- RemotePartyNum
  - getAllCallRmtPtyNum, [230](#)
- remotePartyNum, [658](#)
  - numLen, [659](#)
  - presentationInd, [659](#)
  - remPartyNumber, [659](#)
- ReqFieldsList, [659](#)
  - requestFields, [659](#)
  - requestFieldsLen, [659](#)
- requestFields
  - ReqFieldsList, [659](#)
- requestFieldsLen
  - ReqFieldsList, [659](#)
- resBerRatio
  - LibPackUMTSQoS, [334](#)
  - UMTSMinQoS, [838](#)
  - UMTSQoS, [840](#)
  - wds\_UMTSMinQoS, [1050](#)
- ResCode
  - FirmwareUpdatStat, [222](#)
  - GetAudioVolTLBConfigResp, [236](#)
  - SetAudioVolTLBConfigResp, [688](#)
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [865](#)
- reserved
  - omaDmFotaTlvExt, [518](#)
- resetInfoInd
  - pack\_dms\_SLQSDmsSwiIndicationRegister\_t, [524](#)
- ResetInfoNotification
  - qaGobiApiCbk.h, [1170](#)
- ResetPDSDData
  - qaGobiApiPds.h, [1372](#)
- ResetToFactoryDefaults
  - qaGobiApiDms.h, [1263](#)
- RespFieldsList, [660](#)
  - responseFields, [660](#)
  - responseFieldsLen, [660](#)
- responseFields
  - RespFieldsList, [660](#)
- responseFieldsLen
  - RespFieldsList, [660](#)

- results
  - QmiNasGetRFBandInfoResp, [642](#)
  - QmiNasPerformNetworkScanResp, [643](#)
- RetryCount
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [946](#)
- revPolarity
  - lineCtrlInfo, [336](#)
- revTunneling
  - unpack\_wds\_GetMobileIPProfile\_t, [959](#)
- ReverseMac
  - protocolSubtypeElement, [613](#)
- RmTransferStaticsReq
  - pack\_wds\_RMSetTransferStatistics\_t, [568](#)
- rmTransferStaticsReq, [661](#)
  - bResetStatistics, [661](#)
  - ulMask, [661](#)
- RmtPtyNum
  - arrRemotePartyNum, [113](#)
- roamIndList, [661](#)
  - numInstances, [662](#)
  - radioInterface, [662](#)
  - roamIndicator, [662](#)
- roamIndicator
  - nas\_roamIndList, [443](#)
  - roamIndList, [662](#)
- RoamIndicatorVal
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- roamIndicatorVal
  - qaQmiServingSystemParam, [619](#)
- roamOrigVoiceSO
  - prefVoiceSO, [600](#)
- RoamPref
  - NASRoamPreferenceTlv, [493](#)
- roamStatus
  - nas\_sysInfoCommon, [452](#)
  - sysInfoCommon, [775](#)
- roamStatusValid
  - nas\_sysInfoCommon, [452](#)
  - sysInfoCommon, [775](#)
- roamTimer, [662](#)
  - namID, [663](#)
  - roamTimerValue, [663](#)
- roamTimerValue
  - roamTimer, [663](#)
- Roaming
  - nas\_QmiNas3GppNetworkInfo, [440](#)
  - SlqsNas3GppNetworkInfo, [718](#)
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
  - unpack\_nas\_GetServingNetwork\_t, [888](#)
- roaming
  - unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, [893](#)
- roaming\_ind
  - RoamingInfo, [662](#)
- RoamingIndicatorList
  - qaQmiServingSystemParam, [619](#)
  - unpack\_nas\_SLQSGetServingSystem\_t, [897](#)
- RoamingInfo, [662](#)
  - roaming\_ind, [662](#)
  - TlvPresent, [662](#)
- routeList
  - smsSetRoutesReq, [744](#)
- routeStorage
  - smsRouteEntry, [744](#)
- rptRate
  - LTESigRptCfg, [371](#)
  - LTESigRptConfig, [372](#)
  - nas\_LTESigRptConfig, [426](#)
- rscp
  - nas\_UMTSInfo, [458](#)
  - rxInfo, [668](#)
  - TDSCDMASigInfoExt, [778](#)
  - tdscdmaSigInfoExt, [778](#)
  - UMTSInfo, [834](#)
- rsrp
  - cellParams, [155](#)
  - LTESSInfo, [374](#)
  - lteSSInfo, [375](#)
  - nas\_cellParams, [395](#)
  - nas\_umtsLTENbrCell, [460](#)
  - rxInfo, [668](#)
  - umtsLTENbrCell, [835](#)
- rsrpIlevel
  - lteRsrpInformation, [369](#)
  - nas\_lteRsrpInformation, [424](#)
- rsrq
  - cellParams, [155](#)
  - LTESSInfo, [374](#)
  - lteSSInfo, [375](#)
  - nas\_cellParams, [395](#)
  - nas\_rsrqInformation, [444](#)
  - nas\_umtsLTENbrCell, [460](#)
  - rsrqInformation, [664](#)
  - umtsLTENbrCell, [835](#)
- rsrqDelta
  - nas\_SLQSSignalStrengthsIndReq, [448](#)
  - SLQSSignalStrengthsIndReq, [727](#)
- rsrqInfo
  - nas\_SLQSSignalStrengthsInformation, [449](#)
  - slqsSignalStrengthInfo, [725](#)
  - SLQSSignalStrengthsInformation, [729](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- rsrqInformation, [664](#)
  - radioIf, [664](#)
  - rsrq, [664](#)
- rssi
  - CDMASSInfo, [149](#)
  - cdmaSSInfo, [149](#)
  - cellParams, [155](#)
  - gsmCellInfo, [267](#)
  - HDRSSInfo, [279](#)
  - hdrSSInfo, [280](#)
  - LTESSInfo, [374](#)
  - lteSSInfo, [375](#)
  - nas\_cellParams, [395](#)

- nas\_gsmCellInfo, [406](#)
- TDSCDMASigInfoExt, [778](#)
- tdscdmaSigInfoExt, [778](#)
- unpack\_nas\_GetSignalStrengths\_t, [889](#)
- rts
  - WdsRunTimeSettings, [1065](#)
- rx\_bytes
  - NetStats, [503](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- rx\_errors
  - NetStats, [503](#)
- rx\_overflows
  - NetStats, [503](#)
- rx\_packets
  - NetStats, [503](#)
- rx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [970](#)
- RxDropConutTlv
  - QmiCbkWdsStatisticsIndState, [639](#)
- rxInfo, [667](#)
  - ecio, [668](#)
  - isRadioTuned, [668](#)
  - phase, [668](#)
  - rscp, [668](#)
  - rsrp, [668](#)
  - rxPower, [668](#)
- rxLev
  - GERANInfo, [228](#)
  - nas\_GERANInfo, [404](#)
- rxOKBytesCount
  - unpack\_wds\_SLQSSetDUNCallInfo\_t, [965](#)
- RxOkByteCountTlv
  - QmiCbkWdsStatisticsIndState, [640](#)
- RxOkConutTlv
  - QmiCbkWdsStatisticsIndState, [640](#)
- rxPower
  - rxInfo, [668](#)
- RxQFilter
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- RxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, [922](#)
- rxSignalStrength
  - nas\_rxSignalStrengthListElement, [445](#)
  - rxSignalStrengthListElement, [671](#)
- rxSignalStrengthDelta
  - nas\_SLQSSignalStrengthsIndReq, [448](#)
  - SLQSSignalStrengthsIndReq, [727](#)
- rxSignalStrengthInfo
  - nas\_SLQSSignalStrengthsInformation, [449](#)
  - SLQSSignalStrengthsInformation, [729](#)
- rxSignalStrengthList
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- rxSignalStrengthListElement, [670](#)
  - radiolf, [671](#)
  - rxSignalStrength, [671](#)
- rxSignalStrengthListLen
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- s
  - qmifwinfo\_s, [641](#)
- SMS\_EVENT\_ETWS
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_ETWS\_PLMN
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_MESSAGE\_MODE
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_MT\_MESSAGE
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_SMS\_ON\_IMS
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_SMSC\_ADDRESS
  - qaGobiApiCbk.h, [1200](#)
- SMS\_EVENT\_TRANSFER\_ROUTE\_MT\_MESSAGE
  - qaGobiApiCbk.h, [1200](#)
- sApnExtraParams, [671](#)
  - ambr\_dl, [672](#)
  - ambr\_dl\_ext, [672](#)
  - ambr\_dl\_ext2, [672](#)
  - ambr\_ul, [672](#)
  - ambr\_ul\_ext, [672](#)
  - ambr\_ul\_ext2, [672](#)
  - apnId, [672](#)
- SCI
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- SCM
  - unpack\_nas\_GetCDMANetworkParameters\_t, [885](#)
- SDPTlv
  - NASQmiCbkNasSystemSelPrefInd, [493](#)
- SDU\_HDR\_LEN
  - common.h, [1071](#)
- SECOND\_INSTANCE
  - qaGobiApiCbk.h, [1165](#)
- sGetDeviceSeriesResult, [707](#)
  - eDevice, [707](#)
  - uResult, [707](#)
- SHORT
  - SwiDataTypes.h, [1571](#)
- SI
  - calledPartyInfo, [125](#)
  - callFWExtInfo, [129](#)
  - callingPartyInfo, [133](#)
  - redirNumInfo, [656](#)
- SITlv
  - QmiCbkSwiOmaDmEventStatusReportInd, [638](#)
  - QmiCbkSwiOmaDmEventStatusReportIndExt, [638](#)
- sIntraSearch
  - LTEInfoIntrafreq, [365](#)
  - nas\_LTEInfoIntrafreq, [422](#)
- SLQSAutoConnect
  - qaGobiApiWds.h, [1529](#)
- SLQSCDMADecodeMTTextMsg
  - qaGobiApiSms.h, [1399](#)
- SLQSCDMAEncodeMOTextMsg

- qaGobiApiSms.h, [1399](#)
- SLQSConfigSigInfo
  - qaGobiApiNas.h, [1347](#)
- SLQSCreateProfile
  - qaGobiApiWds.h, [1530](#)
- SLQSDeleteProfile
  - qaGobiApiWds.h, [1530](#)
- SLQSDeleteProfileParams, [715](#)
  - profileIndex, [716](#)
  - profileType, [716](#)
- SLQSDeleteSMS
  - qaGobiApiSms.h, [1400](#)
- SLQSDmsSwiGetResetInfo
  - qaGobiApiDms.h, [1264](#)
- SLQSDmsSwiIndicationRegister
  - qaGobiApiDms.h, [1264](#)
- SLQSDownloadFirmwareToSlot
  - qaGobiApiFms.h, [1292](#)
- SLQSFWINFO\_SKU\_SZ
  - qaGobiApiFms.h, [1287](#)
- SLQSGet3GPPConfigItem
  - qaGobiApiWds.h, [1531](#)
- SLQSGetAGPSCfg
  - qaGobiApiPds.h, [1376](#)
- SLQSGetAudioPathConfig
  - qaGobiApiAudio.h, [1152](#)
- SLQSGetAudioProfile
  - qaGobiApiAudio.h, [1152](#)
- SLQSGetAudioVolTLBConfig
  - qaGobiApiAudio.h, [1153](#)
- SLQSGetBandCapabilities
  - qaGobiApiDms.h, [1264](#)
- SLQSGetBandCapability
  - qaGobiApiDms.h, [1264](#)
- SLQSGetBootVersionNumber
  - qaGobiApiFms.h, [1293](#)
- SLQSGetByteTotals
  - qaGobiApiWds.h, [1531](#)
- SLQSGetConnectionRate
  - qaGobiApiWds.h, [1532](#)
- SLQSGetCurrDataSystemStat
  - qaGobiApiWds.h, [1532](#)
- SLQSGetCurrentChannelRate
  - qaGobiApiWds.h, [1533](#)
- SLQSGetCurrentPRLInfo
  - qaGobiApiDms.h, [1266](#)
- SLQSGetCustFeatures
  - qaGobiApiDms.h, [1266](#)
- SLQSGetCustFeaturesV2
  - qaGobiApiDms.h, [1267](#)
- SLQSGetDUNCallInfo
  - qaGobiApiWds.h, [1534](#)
- SLQSGetDataBearerTechnology
  - qaGobiApiWds.h, [1533](#)
- SLQSGetDataBearerTechnologyExt
  - qaGobiApiWds.h, [1534](#)
- SLQSGetDeviceMode
  - qaGobiApiDcs.h, [1241](#)
- SLQSGetERIFile
  - qaGobiApiDms.h, [1267](#)
- SLQSGetErrorRate
  - qaGobiApiNas.h, [1348](#)
- SLQSGetFirmwareInfo
  - qaGobiApiFms.h, [1294](#)
- SLQSGetGPSStateInfo
  - qaGobiApiPds.h, [1376](#)
- SLQSGetIMSARegStatus
  - qaGobiApiImsa.h, [1307](#)
- SLQSGetIMSAServiceStatus
  - qaGobiApiImsa.h, [1308](#)
- SLQSGetIMSASupportedFields
  - qaGobiApiImsa.h, [1308](#)
- SLQSGetIMSASupportedMsg
  - qaGobiApiImsa.h, [1309](#)
- SLQSGetIMSSMSCfg
  - qaGobiApiIms.h, [1301](#)
- SLQSGetIMSUserConfig
  - qaGobiApiIms.h, [1302](#)
- SLQSGetIMSVolPConfig
  - qaGobiApiIms.h, [1302](#)
- SLQSGetImageInfo
  - qaGobiApiFms.h, [1294](#)
- SLQSGetImageInfo\_9x15
  - qaGobiApiFms.h, [1295](#)
- SLQSGetImageInfoMC77xx
  - qaGobiApiFms.h, [1295](#)
- SLQSGetImageInfoMC83xx
  - qaGobiApiFms.h, [1296](#)
- SLQSGetIndicationRegister
  - qaGobiApiSms.h, [1401](#)
- SLQSGetM2MAVMute
  - qaGobiApiSwiAudio.h, [1416](#)
- SLQSGetM2MAudioProfile
  - qaGobiApiSwiAudio.h, [1415](#)
- SLQSGetM2MAudioVolume
  - qaGobiApiSwiAudio.h, [1416](#)
- SLQSGetM2MSpkrGain
  - qaGobiApiSwiAudio.h, [1416](#)
- SLQSGetMessageWaiting
  - qaGobiApiSms.h, [1401](#)
- SLQSGetNetStatistic
  - qaGobiApiDcs.h, [1241](#)
- SLQSGetNetworkTime
  - qaGobiApiNas.h, [1348](#)
- SLQSGetOperatorNameData
  - qaGobiApiNas.h, [1348](#)
- SLQSGetPLMNNName
  - qaGobiApiNas.h, [1349](#)
- SLQSGetPacketStatistics
  - qaGobiApiWds.h, [1534](#)
- SLQSGetPidof
  - qaGobiApiSwi.h, [1413](#)
- SLQSGetProfile
  - qaGobiApiWds.h, [1535](#)
- SLQSGetProfileSettings
  - qaGobiApiWds.h, [1536](#)

- SLQSGetRegMgrConfig
  - qaGobiApiIms.h, [1303](#)
- SLQSGetRfSarState
  - qaGobiApiSar.h, [1390](#)
- SLQSGetRuntimeSettings
  - qaGobiApiWds.h, [1537](#)
- SLQSGetSIPConfig
  - qaGobiApiIms.h, [1303](#)
- SLQSGetSMS
  - qaGobiApiSms.h, [1402](#)
- SLQSGetSMSList
  - qaGobiApiSms.h, [1403](#)
- SLQSGetSdkVersion
  - qaGobiApiSwi.h, [1414](#)
- SLQSGetSerialNumbers
  - qaGobiApiDms.h, [1267](#)
- SLQSGetServingSystem
  - qaGobiApiNas.h, [1349](#)
- SLQSGetSessionState
  - qaGobiApiWds.h, [1537](#)
- SLQSGetSignalStrength
  - qaGobiApiNas.h, [1350](#)
- SLQSGetSmsBroadcastConfig
  - qaGobiApiSms.h, [1403](#)
- SLQSGetSysSelectionPref
  - qaGobiApiNas.h, [1350](#)
- SLQSGetTransLayerInfo
  - qaGobiApiSms.h, [1404](#)
- SLQSGetTransNWRglInfo
  - qaGobiApiSms.h, [1405](#)
- SLQSGetUsbPortNames
  - qaGobiApiDcs.h, [1242](#)
- SLQSGetValidFwPriCombinations
  - qaGobiApiFms.h, [1297](#)
- SLQSImsConfigIndicationRegister
  - qaGobiApiIms.h, [1303](#)
- SLQSInitiateNetworkRegistration
  - qaGobiApiNas.h, [1351](#)
- SLQSIspkgFormatRequired
  - qaGobiApiFms.h, [1297](#)
- SLQSKillSDKProcess
  - qaGobiApiDcs.h, [1242](#)
- SLQSLOCDeAssData
  - qaGobiApiLoc.h, [1311](#)
- SLQSLOCEventRegister
  - qaGobiApiLoc.h, [1311](#)
- SLQSLOCGetBestAvailPos
  - qaGobiApiLoc.h, [1312](#)
- SLQSLOCInjectPosition
  - qaGobiApiLoc.h, [1312](#)
- SLQSLOCInjectSensorData
  - qaGobiApiLoc.h, [1312](#)
- SLQSLOCInjectUTCTime
  - qaGobiApiLoc.h, [1313](#)
- SLQSLOCSetCradleMountConfig
  - qaGobiApiLoc.h, [1313](#)
- SLQSLOCSetExtPowerState
  - qaGobiApiLoc.h, [1314](#)
- SLQSLOCSetOpMode
  - qaGobiApiLoc.h, [1314](#)
- SLQSLOCStart
  - qaGobiApiLoc.h, [1315](#)
- SLQSLOCStop
  - qaGobiApiLoc.h, [1315](#)
- SLQSModifyProfile
  - qaGobiApiWds.h, [1538](#)
- SLQSModifySMSStatus
  - qaGobiApiSms.h, [1405](#)
- SLQSNASGetLTEPHYCaInfo
  - qaGobiApiNas.h, [1353](#)
- SLQSNASSwiGetChannelLock
  - qaGobiApiNas.h, [1356](#)
- SLQSNASSwiSetChannelLock
  - qaGobiApiNas.h, [1358](#)
- SLQSNasConfigSigInfo2
  - qaGobiApiNas.h, [1351](#)
- SLQSNasGet3GPP2Subscription
  - qaGobiApiNas.h, [1352](#)
- SLQSNasGetCellLocationInfo
  - qaGobiApiNas.h, [1352](#)
- SLQSNasGetHDRColorCode
  - qaGobiApiNas.h, [1353](#)
- SLQSNasGetSigInfo
  - qaGobiApiNas.h, [1353](#)
- SLQSNasGetSysInfo
  - qaGobiApiNas.h, [1354](#)
- SLQSNasGetTxRxInfo
  - qaGobiApiNas.h, [1354](#)
- SLQSNasIndicationRegister
  - qaGobiApiNas.h, [1355](#)
- SLQSNasIndicationRegisterExt
  - qaGobiApiNas.h, [1356](#)
- SLQSNasIndicationRegisterLTECphyCa
  - qaGobiApiNas.h, [1356](#)
- SLQSNasNetworkTimeCallBack
  - qaGobiApiCbk.h, [1215](#)
- SLQSNasSigInfo2CallBack
  - qaGobiApiCbk.h, [1215](#)
- SLQSNasSigInfoCallBack
  - qaGobiApiCbk.h, [1216](#)
- SLQSNasSwiIndicationRegister
  - qaGobiApiNas.h, [1357](#)
- SLQSNasSwiModemStatus
  - qaGobiApiNas.h, [1357](#)
- SLQSNasSwiOTAMessageCallback
  - qaGobiApiCbk.h, [1217](#)
- SLQSNasSysInfoCallBack
  - qaGobiApiCbk.h, [1217](#)
- SLQSOMADMCancelSession
  - qaGobiApiSwiOmadms.h, [1425](#)
- SLQSOMADMGetSessionInfo
  - qaGobiApiSwiOmadms.h, [1425](#)
- SLQSOMADMGetSettings
  - qaGobiApiSwiOmadms.h, [1426](#)
- SLQSOMADMGetSettings2
  - qaGobiApiSwiOmadms.h, [1426](#)

- SLQSOMADMSendSelection
  - qaGobiApiSwiOmadms.h, [1427](#)
- SLQSOMADMSendSelection2
  - qaGobiApiSwiOmadms.h, [1427](#)
- SLQSOMADMSessionInfo
  - qaGobiApiSwiOmadms.h, [1421](#)
- SLQSOMADMSetSettings
  - qaGobiApiSwiOmadms.h, [1428](#)
- SLQSOMADMSetSettings2
  - qaGobiApiSwiOmadms.h, [1429](#)
- SLQSOMADMSetSettings3
  - qaGobiApiSwiOmadms.h, [1429](#)
- SLQSOMADMSettings
  - qaGobiApiSwiOmadms.h, [1422](#)
- SLQSOMADMSettingsReqParams
  - qaGobiApiSwiOmadms.h, [1423](#)
- SLQSOMADMSettingsReqParams3
  - qaGobiApiSwiOmadms.h, [1424](#)
- SLQSOMADMStartSession
  - qaGobiApiSwiOmadms.h, [1429](#)
- SLQSOMADMStartSession2
  - qaGobiApiSwiOmadms.h, [1430](#)
- SLQSOriginateUSSD
  - qaGobiApiVoice.h, [1487](#)
- SLQSPDSDeterminePosition
  - qaGobiApiPds.h, [1377](#)
- SLQSPDSInjectAbsoluteTimeReference
  - qaGobiApiPds.h, [1377](#)
- SLQSPDSInjectPositionData
  - qaGobiApiPds.h, [1378](#)
- SLQSPerformNetworkScan
  - qaGobiApiNas.h, [1358](#)
- SLQSQosGetFlowStatus
  - qaGobiApiQos.h, [1382](#)
- SLQSQosGetGranted
  - qaGobiApiQos.h, [1382](#)
- SLQSQosGetNWProf
  - qaGobiApiQos.h, [1383](#)
- SLQSQosGetNetworkStatus
  - qaGobiApiQos.h, [1383](#)
- SLQSQosModify
  - qaGobiApiQos.h, [1383](#)
- SLQSQosRel
  - qaGobiApiQos.h, [1384](#)
- SLQSQosReq
  - qaGobiApiQos.h, [1384](#)
- SLQSQosReset
  - qaGobiApiQos.h, [1385](#)
- SLQSQosResume
  - qaGobiApiQos.h, [1385](#)
- SLQSQosSuspend
  - qaGobiApiQos.h, [1386](#)
- SLQSQosSwiReadApnExtraParams
  - qaGobiApiQos.h, [1386](#)
- SLQSQosSwiReadDataStats
  - qaGobiApiQos.h, [1387](#)
- SLQSRegisterIMSAIndication
  - qaGobiApiImsa.h, [1309](#)
- SLQSResetPacketStatics
  - qaGobiApiWds.h, [1539](#)
- SLQSSetDHCIPv4ClientConfig
  - qaGobiApiWds.h, [1541](#)
- SLQSSetLoopback
  - qaGobiApiWds.h, [1541](#)
- SLQSSSTlv
  - unpack\_nas\_SetEventReportInd\_t, [891](#)
- SLQSSetDHCIPv4ClientConfig
  - qaGobiApiWds.h, [1541](#)
- SLQSSetLoopback
  - qaGobiApiWds.h, [1542](#)
- SLQSSendAsyncSMS
  - qaGobiApiSms.h, [1406](#)
- SLQSSendLongSMS
  - qaGobiApiSms.h, [1406](#)
- SLQSSendRawQMI
  - qaGobiApiSwi.h, [1414](#)
- SLQSSendSMS
  - qaGobiApiSms.h, [1407](#)
- SLQSSet3GPPConfigItem
  - qaGobiApiWds.h, [1539](#)
- SLQSSetAGPSConfig
  - qaGobiApiPds.h, [1378](#)
- SLQSSetAudioPathConfig
  - qaGobiApiAudio.h, [1153](#)
- SLQSSetAudioProfile
  - qaGobiApiAudio.h, [1154](#)
- SLQSSetAudioVolTLBConfig
  - qaGobiApiAudio.h, [1154](#)
- SLQSSetBandPreference
  - qaGobiApiNas.h, [1358](#)
- SLQSSetBandPreferenceCbk
  - qaGobiApiCbk.h, [1218](#)
- SLQSSetCustFeatures
  - qaGobiApiDms.h, [1268](#)
- SLQSSetCustFeaturesV2
  - qaGobiApiDms.h, [1268](#)
- SLQSSetDHCIPv4ClientLeaseStatusCallback
  - qaGobiApiCbk.h, [1218](#)
- SLQSSetDUNCallInfoCallback
  - qaGobiApiCbk.h, [1219](#)
- SLQSSetDataSystemStatusCallback
  - qaGobiApiCbk.h, [1218](#)
- SLQSSetIMSAPdpStatusCallback
  - qaGobiApiCbk.h, [1219](#)
- SLQSSetIMSARegStatusCallback
  - qaGobiApiCbk.h, [1220](#)
- SLQSSetIMSARegStatusCallback
  - qaGobiApiCbk.h, [1220](#)
- SLQSSetIMSASvcStatusCallback
  - qaGobiApiCbk.h, [1220](#)
- SLQSSetIMSSMSCConfig
  - qaGobiApiImms.h, [1304](#)
- SLQSSetIMSSMSCConfigCallback
  - qaGobiApiCbk.h, [1221](#)
- SLQSSetIMSUserConfig
  - qaGobiApiImms.h, [1304](#)

- SLQSSetIMSUserConfigCallback  
qaGobiApiCbk.h, [1221](#)
- SLQSSetIMSVoIPConfig  
qaGobiApiIms.h, [1305](#)
- SLQSSetIMSVoIPConfigCallback  
qaGobiApiCbk.h, [1222](#)
- SLQSSetIndicationRegister  
qaGobiApiSms.h, [1408](#)
- SLQSSetLocInjectPositionCallback  
qaGobiApiCbk.h, [1222](#)
- SLQSSetLocInjectUTCTimeCallback  
qaGobiApiCbk.h, [1222](#)
- SLQSSetLoggingMask  
qaGobiApiDcs.h, [1243](#)
- SLQSSetM2MAVMute  
qaGobiApiSwtAudio.h, [1419](#)
- SLQSSetM2MAudioAVCFG  
qaGobiApiSwtAudio.h, [1417](#)
- SLQSSetM2MAudioLPBK  
qaGobiApiSwtAudio.h, [1417](#)
- SLQSSetM2MAudioNVDef  
qaGobiApiSwtAudio.h, [1418](#)
- SLQSSetM2MAudioProfile  
qaGobiApiSwtAudio.h, [1418](#)
- SLQSSetM2MAudioVolume  
qaGobiApiSwtAudio.h, [1418](#)
- SLQSSetM2MSpkrgain  
qaGobiApiSwtAudio.h, [1419](#)
- SLQSSetModemTempCallback  
qaGobiApiCbk.h, [1223](#)
- SLQSSetPacketSrvStatusCallback  
qaGobiApiCbk.h, [1223](#)
- SLQSSetPositionMethodState  
qaGobiApiPds.h, [1379](#)
- SLQSSetProfile  
qaGobiApiWds.h, [1539](#)
- SLQSSetQosEventCallback  
qaGobiApiCbk.h, [1223](#)
- SLQSSetQosNWStatusCallback  
qaGobiApiCbk.h, [1224](#)
- SLQSSetQosPriEventCallback  
qaGobiApiCbk.h, [1224](#)
- SLQSSetQosStatusCallback  
qaGobiApiCbk.h, [1225](#)
- SLQSSetRegMgrConfig  
qaGobiApiIms.h, [1305](#)
- SLQSSetRegMgrConfigCallback  
qaGobiApiCbk.h, [1225](#)
- SLQSSetRfSarState  
qaGobiApiSar.h, [1390](#)
- SLQSSetSDKTerminatedCallback  
qaGobiApiCbk.h, [1226](#)
- SLQSSetSIMBasedImageSwitching  
qaGobiApiFms.h, [1297](#)
- SLQSSetSIPConfig  
qaGobiApiIms.h, [1306](#)
- SLQSSetSIPConfigCallback  
qaGobiApiCbk.h, [1228](#)
- SLQSSetSMSEventCallback  
qaGobiApiCbk.h, [1228](#)
- SLQSSetServingSystemCallback  
qaGobiApiCbk.h, [1226](#)
- SLQSSetSessionStateCallback  
qaGobiApiCbk.h, [1227](#)
- SLQSSetSignalStrengthsCallback  
qaGobiApiCbk.h, [1227](#)
- SLQSSetSmsBroadcastActivation  
qaGobiApiSms.h, [1408](#)
- SLQSSetSmsBroadcastConfig  
qaGobiApiSms.h, [1409](#)
- SLQSSetSmsStorage  
qaGobiApiSms.h, [1409](#)
- SLQSSetSpkgFormatRequired  
qaGobiApiFms.h, [1298](#)
- SLQSSetSwtGetResetInfoCallback  
qaGobiApiCbk.h, [1228](#)
- SLQSSetSwtHDRPersCallback  
qaGobiApiCbk.h, [1228](#)
- SLQSSetSysSelectionPref  
qaGobiApiNas.h, [1360](#)
- SLQSSetSysSelectionPrefCallback  
qaGobiApiCbk.h, [1229](#)
- SLQSSetTransLayerInfoCallback  
qaGobiApiCbk.h, [1229](#)
- SLQSSetTransNWRegInfoCallback  
qaGobiApiCbk.h, [1230](#)
- SLQSSetWdsEventCallback  
qaGobiApiCbk.h, [1230](#)
- SLQSSetWdsTransferStatisticCallback  
qaGobiApiCbk.h, [1231](#)
- SLQSSignalStrengthsIndReq, [726](#)  
ecioDelta, [727](#)  
ecioThresholdList, [727](#)  
ecioThresholdListLen, [727](#)  
ioDelta, [727](#)  
lteRsrpDelta, [727](#)  
lteSnrDelta, [727](#)  
rsrqDelta, [727](#)  
rxSignalStrengthDelta, [727](#)  
sinrDelta, [727](#)  
sinrThresholdList, [727](#)  
sinrThresholdListLen, [727](#)
- SLQSSignalStrengthsInformation, [728](#)  
eciInfo, [728](#)  
errorRateInfo, [729](#)  
io, [729](#)  
lteRsrpinfo, [729](#)  
lteSnrinfo, [729](#)  
rsrqInfo, [729](#)  
rxSignalStrengthInfo, [729](#)  
sinr, [729](#)
- SLQSSmsGetMaxStorageSize  
qaGobiApiSms.h, [1410](#)
- SLQSSmsGetMessageProtocol  
qaGobiApiSms.h, [1410](#)
- SLQSSmsSetRoutes

- qaGobiApiSms.h, [1411](#)
- SLQSSStart
  - qaGobiApiDcs.h, [1243](#)
- SLQSSStart\_AVAgent
  - qaGobiApiDcs.h, [1244](#)
- SLQSSStartSrv
  - qaGobiApiDcs.h, [1244](#)
- SLQSSStartStopDataSession
  - qaGobiApiWds.h, [1542](#)
- SLQSSwiClearDyingGaspStatistics
  - qaGobiApiDms.h, [1269](#)
- SLQSSwiGetAllCarrierImages
  - qaGobiApiFms.h, [1298](#)
- SLQSSwiGetCrashAction
  - qaGobiApiDms.h, [1269](#)
- SLQSSwiGetCrashInfo
  - qaGobiApiDms.h, [1269](#)
- SLQSSwiGetDyingGaspCfg
  - qaGobiApiDms.h, [1270](#)
- SLQSSwiGetDyingGaspStatistics
  - qaGobiApiDms.h, [1270](#)
- SLQSSwiGetFSN
  - qaGobiApiDms.h, [1271](#)
- SLQSSwiGetFirmwareCurr
  - qaGobiApiDms.h, [1270](#)
- SLQSSwiGetFwUpdateStatus
  - qaGobiApiDms.h, [1271](#)
- SLQSSwiGetHDRPersonality
  - qaGobiApiNas.h, [1360](#)
- SLQSSwiGetHDRProtSubtype
  - qaGobiApiNas.h, [1361](#)
- SLQSSwiGetHRPDStats
  - qaGobiApiNas.h, [1361](#)
- SLQSSwiGetHostDevInfo
  - qaGobiApiDms.h, [1272](#)
- SLQSSwiGetHostDevInfoParams
  - qaGobiApiDms.h, [1252](#)
- SLQSSwiGetLteCQI
  - qaGobiApiNas.h, [1361](#)
- SLQSSwiGetOSInfo
  - qaGobiApiDms.h, [1272](#)
- SLQSSwiGetOSInfoParams
  - qaGobiApiDms.h, [1252](#)
- SLQSSwiGetSMSStorage
  - qaGobiApiSms.h, [1411](#)
- SLQSSwiGetSerialNoExt
  - qaGobiApiDms.h, [1272](#)
- SLQSSwiGetSerialNoExtParams
  - qaGobiApiDms.h, [1253](#)
- SLQSSwiGetUSBComp
  - qaGobiApiDms.h, [1273](#)
- SLQSSwiNetworkDebug
  - qaGobiApiNas.h, [1362](#)
- SLQSSwiPSDetach
  - qaGobiApiNas.h, [1362](#)
- SLQSSwiSetCrashAction
  - qaGobiApiDms.h, [1273](#)
- SLQSSwiSetDyingGaspCfg
  - qaGobiApiDms.h, [1274](#)
- SLQSSwiSetHostDevInfo
  - qaGobiApiDms.h, [1274](#)
- SLQSSwiSetHostDevInfoParams
  - qaGobiApiDms.h, [1253](#)
- SLQSSwiSetOSInfo
  - qaGobiApiDms.h, [1275](#)
- SLQSSwiSetOSInfoParams
  - qaGobiApiDms.h, [1254](#)
- SLQSSwiSetUSBComp
  - qaGobiApiDms.h, [1275](#)
- SLQSTmdDeRegNotMitigationLvl
  - qaGobiApiTmd.h, [1468](#)
- SLQSTmdGetMitigationDevList
  - qaGobiApiTmd.h, [1468](#)
- SLQSTmdGetMitigationLvl
  - qaGobiApiTmd.h, [1469](#)
- SLQSTmdMitigationLvlRptCallback
  - qaGobiApiCbk.h, [1232](#)
- SLQSTmdRegNotMitigationLvl
  - qaGobiApiTmd.h, [1469](#)
- SLQSUIMAuthenticate
  - qaGobiApiUim.h, [1472](#)
- SLQSUIMChangePin
  - qaGobiApiUim.h, [1472](#)
- SLQSUIMDepersonalization
  - qaGobiApiUim.h, [1473](#)
- SLQSUIMEventRegister
  - qaGobiApiUim.h, [1473](#)
- SLQSUIMGetCardStatus
  - qaGobiApiUim.h, [1474](#)
- SLQSUIMGetConfiguration
  - qaGobiApiUim.h, [1474](#)
- SLQSUIMGetFileAttributes
  - qaGobiApiUim.h, [1475](#)
- SLQSUIMGetSlotsStatus
  - qaGobiApiUim.h, [1476](#)
- SLQSUIMGetState
  - qaGobiApiDms.h, [1275](#)
- SLQSUIMPowerDown
  - qaGobiApiUim.h, [1476](#)
- SLQSUIMPowerUp
  - qaGobiApiUim.h, [1476](#)
- SLQSUIMReadTransparent
  - qaGobiApiUim.h, [1477](#)
- SLQSUIMRefreshComplete
  - qaGobiApiUim.h, [1477](#)
- SLQSUIMRefreshGetLastEvent
  - qaGobiApiUim.h, [1478](#)
- SLQSUIMRefreshOK
  - qaGobiApiUim.h, [1478](#)
- SLQSUIMRefreshRegister
  - qaGobiApiUim.h, [1479](#)
- SLQSUIMReset
  - qaGobiApiUim.h, [1479](#)
- SLQSUIMSetPinProtection
  - qaGobiApiUim.h, [1480](#)
- SLQSUIMSetRefreshCallBack

- qaGobiApiCbk.h, [1232](#)
- SLQSUIMSetStatusChangeCallBack
  - qaGobiApiCbk.h, [1232](#)
- SLQSUIMSwitchSlot
  - qaGobiApiUim.h, [1480](#)
- SLQSUIMUnblockPin
  - qaGobiApiUim.h, [1481](#)
- SLQSUIMVerifyPin
  - qaGobiApiUim.h, [1481](#)
- SLQSupgradeFirmware9x15
  - qaGobiApiFms.h, [1299](#)
- SLQSVoiceALSSelectLine
  - qaGobiApiVoice.h, [1487](#)
- SLQSVoiceALSSetLineSwitching
  - qaGobiApiVoice.h, [1488](#)
- SLQSVoiceAnswerCall
  - qaGobiApiVoice.h, [1488](#)
- SLQSVoiceBindSubscription
  - qaGobiApiVoice.h, [1489](#)
- SLQSVoiceBurstDTMF
  - qaGobiApiVoice.h, [1489](#)
- SLQSVoiceDialCall
  - qaGobiApiVoice.h, [1490](#)
- SLQSVoiceEndCall
  - qaGobiApiVoice.h, [1490](#)
- SLQSVoiceGetAllCallInfo
  - qaGobiApiVoice.h, [1491](#)
- SLQSVoiceGetCLIP
  - qaGobiApiVoice.h, [1493](#)
- SLQSVoiceGetCLIR
  - qaGobiApiVoice.h, [1494](#)
- SLQSVoiceGetCNAP
  - qaGobiApiVoice.h, [1494](#)
- SLQSVoiceGetCOLP
  - qaGobiApiVoice.h, [1495](#)
- SLQSVoiceGetCOLR
  - qaGobiApiVoice.h, [1495](#)
- SLQSVoiceGetCallBarring
  - qaGobiApiVoice.h, [1491](#)
- SLQSVoiceGetCallForwardingStatus
  - qaGobiApiVoice.h, [1492](#)
- SLQSVoiceGetCallInfo
  - qaGobiApiVoice.h, [1492](#)
- SLQSVoiceGetCallWaiting
  - qaGobiApiVoice.h, [1493](#)
- SLQSVoiceGetConfig
  - qaGobiApiVoice.h, [1496](#)
- SLQSVoiceIndicationRegister
  - qaGobiApiVoice.h, [1497](#)
- SLQSVoiceInfoRecCallback
  - qaGobiApiCbk.h, [1233](#)
- SLQSVoiceManageCalls
  - qaGobiApiVoice.h, [1497](#)
- SLQSVoiceOrigUSSDNoWait
  - qaGobiApiVoice.h, [1498](#)
- SLQSVoiceSendFlash
  - qaGobiApiVoice.h, [1498](#)
- SLQSVoiceSetAllCallStatusCallBack
  - qaGobiApiCbk.h, [1233](#)
- SLQSVoiceSetCallBarringPassword
  - qaGobiApiVoice.h, [1499](#)
- SLQSVoiceSetConfig
  - qaGobiApiVoice.h, [1499](#)
- SLQSVoiceSetDTMFEventCallBack
  - qaGobiApiCbk.h, [1233](#)
- SLQSVoiceSetOTASPStatusCallBack
  - qaGobiApiCbk.h, [1234](#)
- SLQSVoiceSetPreferredPrivacy
  - qaGobiApiVoice.h, [1500](#)
- SLQSVoiceSetPrivacyChangeCallBack
  - qaGobiApiCbk.h, [1234](#)
- SLQSVoiceSetSUPSCallBack
  - qaGobiApiCbk.h, [1235](#)
- SLQSVoiceSetSUPSNotificationCallback
  - qaGobiApiCbk.h, [1235](#)
- SLQSVoiceSetSUPSService
  - qaGobiApiVoice.h, [1500](#)
- SLQSVoiceStartContDTMF
  - qaGobiApiVoice.h, [1501](#)
- SLQSVoiceStopContDTMF
  - qaGobiApiVoice.h, [1501](#)
- SLQSWCDMADecodeLongTextMsg
  - qaGobiApiSms.h, [1412](#)
- SLQSWCDMADecodeMTTextMsg
  - qaGobiApiSms.h, [1412](#)
- SLQSWCDMAEncodeMOTextMsg
  - qaGobiApiSms.h, [1412](#)
- SLQSWdsGoActive
  - qaGobiApiWds.h, [1542](#)
- SLQSWdsGoDormant
  - qaGobiApiWds.h, [1543](#)
- SLQSWdsSetEventReport
  - qaGobiApiWds.h, [1543](#)
- SLQSWdsSwiPDPRuntimeSettings
  - qaGobiApiWds.h, [1544](#)
- SLQSWmsAsyncRawSendCallBack
  - qaGobiApiCbk.h, [1236](#)
- SLQSWmsMemoryFullCallBack
  - qaGobiApiCbk.h, [1236](#)
- SLQSWmsMessageWaitingCallBack
  - qaGobiApiCbk.h, [1236](#)
- SMSAsyncRawSend
  - qaGobiApiCbk.h, [1172](#)
- SMSAsyncRawSend\_s, [731](#)
  - alphaIDLen, [732](#)
  - causeCode, [732](#)
  - errorClass, [732](#)
  - messageID, [732](#)
  - msgDelFailureCause, [732](#)
  - msgDelFailureType, [732](#)
  - pAlphaID, [732](#)
  - RPCause, [732](#)
  - sendStatus, [732](#)
  - TPCause, [732](#)
  - userData, [732](#)
- SMSCAddress, [733](#)

- data, 733
- length, 733
- sMSCAddress, 732
  - data, 733
  - length, 733
- SMSCAddressInfo
  - qaGobiApiCbK.h, 1173
- sMSCAddressInfo
  - sms.h, 1563
- sMSCAddressTlv, 733
  - SMSCInfo, 734
  - TlvPresent, 734
- SMSCInfo
  - sMSCAddressTlv, 734
- SMSCtlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 939
- SMSEtwsMessage, 734
  - data, 735
  - length, 735
  - notificationType, 735
- sMSEtwsMessage, 734
  - data, 734
  - length, 734
  - notificationType, 734
- SMSEtwsMessageInfo
  - qaGobiApiCbK.h, 1173
- sMSEtwsMessageInfo
  - sms.h, 1563
- sMSEtwsMessageTlv, 735
  - EtwsMessageInfo, 735
  - TlvPresent, 735
- SMSEtwsPlmn, 736
  - mobileCountryCode, 736
  - mobileNetworkCode, 736
- sMSEtwsPlmn, 735
  - mobileCountryCode, 736
  - mobileNetworkCode, 736
- SMSEtwsPlmnInfo
  - qaGobiApiCbK.h, 1173
- sMSEtwsPlmnInfo
  - sms.h, 1564
- SMSEventInfo
  - qaGobiApiCbK.h, 1173
- SMSEventInfo\_s, 736
  - pEtwsMessageInfo, 737
  - pEtwsPlmnInfo, 737
  - pMTMessageInfo, 738
  - pMessageModelInfo, 737
  - pSMSCAddressInfo, 738
  - pSMSOnIMSInfo, 738
  - pTransferRouteMTMessageInfo, 738
  - smsEventType, 738
- SMSEventType
  - qaGobiApiCbK.h, 1200
- SMSMTMessage, 741
  - messageIndex, 741
  - storageType, 741
- sMSMTMessage, 741
  - messageIndex, 741
  - storageType, 741
- SMSMTMessageInfo
  - qaGobiApiCbK.h, 1174
- sMSMTMessageInfo
  - sms.h, 1564
- SMSMemoryInfo, 739
  - messageMode, 739
  - storageType, 739
- SMSMessageMode, 740
  - messageMode, 740
- sMSMessageMode, 739
  - messageMode, 740
- SMSMessageModelInfo
  - qaGobiApiCbK.h, 1174
- sMSMessageModelInfo
  - sms.h, 1564
- SMSOnIMS, 742
  - smsOnIMS, 742
- sMSOnIMS, 742
  - smsOnIMS, 742
- SMSOnIMSInfo
  - qaGobiApiCbK.h, 1175
- sMSOnIMSInfo
  - sms.h, 1564
- sMSOnIMSTlv, 742
  - IMSInfo, 743
  - TlvPresent, 743
- SMSSupport
  - pack\_dms\_SetCustFeature\_t, 522
  - unpack\_dms\_GetCustFeature\_t, 845
- SMSTransferRouteMTMessage, 745
  - ackIndicator, 746
  - data, 746
  - format, 746
  - length, 746
  - transactionID, 746
- sMSTransferRouteMTMessage, 745
  - ackIndicator, 745
  - data, 745
  - format, 745
  - length, 745
  - transactionID, 745
- SMSTransferRouteMTMessageInfo
  - qaGobiApiCbK.h, 1175
- sMSTransferRouteMTMessageInfo
  - sms.h, 1564
- SNR
  - NetworkStatEVDO, 508
- sNonIntraSearch
  - LTEInfoIntrafreq, 365
  - nas\_LTEInfoIntrafreq, 422
- SO
  - NetworkStat1x, 506
- SOMask
  - CurrNetworkInfo, 178
  - currNetworkInfo, 179
  - wds\_currNetworkInfo, 1043

- sPhyCaAggPcellInfo
  - nasGetLTECphyCa, [471](#)
  - QmiCbkNasLTECphyCaInfo, [638](#)
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [891](#)
- sPhyCaAggScellDIBw
  - nasGetLTECphyCa, [471](#)
  - QmiCbkNasLTECphyCaInfo, [638](#)
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [891](#)
- sPhyCaAggScellIndType
  - nasGetLTECphyCa, [471](#)
  - QmiCbkNasLTECphyCaInfo, [638](#)
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [892](#)
- sPhyCaAggScellIndex
  - nasGetLTECphyCa, [471](#)
  - QmiCbkNasLTECphyCaInfo, [638](#)
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [892](#)
- sPhyCaAggScellInfo
  - nasGetLTECphyCa, [471](#)
  - QmiCbkNasLTECphyCaInfo, [638](#)
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [892](#)
- sQosFlowStat, [746](#)
  - bearerId, [747](#)
  - tx\_bytes, [747](#)
  - tx\_bytes\_drp, [747](#)
  - tx\_pkt, [747](#)
  - tx\_pkt\_drp, [747](#)
- sQosStat, [747](#)
  - apnId, [748](#)
  - numQosFlow, [748](#)
  - qosFlow, [748](#)
  - total\_rx\_bytes, [748](#)
  - total\_rx\_pkt, [748](#)
  - total\_tx\_bytes, [748](#)
  - total\_tx\_bytes\_drp, [748](#)
  - total\_tx\_pkt, [748](#)
  - total\_tx\_pkt\_drp, [748](#)
- SSInfo
  - unpack\_nas\_SetServingSystemCallback\_ind\_t, [893](#)
- sSLQSSignalStrengthsInfo
  - nas\_SLQSSignalStrengthsTlv, [449](#)
- SSTlv
  - unpack\_nas\_SetEventReportInd\_t, [891](#)
- SUPSInfo, [752](#)
  - isModByCC, [753](#)
  - svcType, [753](#)
- SUPSInformation
  - voiceSUPSInfo, [1029](#)
- SUPSType
  - voiceManageCallsReq, [1012](#)
- SV, [753](#)
  - id, [754](#)
  - mask, [754](#)
  - system, [754](#)
- SVInfo, [754](#)
  - len, [754](#)
  - pSV, [755](#)
- SWI Audio Service(SWIAUDIO), [50](#)
- SWI Open Mobile Alliance Service (SWIOMA), [43](#)
- SWI\_API
  - SwiDataTypes.h, [1571](#)
- SWI\_STRUCT\_CarrierImage, [755](#)
  - m\_FwBuildId, [756](#)
  - m\_FwImageId, [756](#)
  - m\_PriBuildId, [756](#)
  - m\_PriImageId, [756](#)
  - m\_nCarrierId, [756](#)
  - m\_nFolderId, [756](#)
  - m\_nStorage, [756](#)
- SWIWWANCMAPI.h, [1582](#)
- samplesPerBatch
  - accelAcceptReady\_s, [93](#)
  - accelTempAcceptReady\_s, [94](#)
  - gyroAcceptReady\_s, [273](#)
  - gyroTempAcceptReady\_s, [273](#)
- satelliteInfo, [672](#)
  - azimuth, [674](#)
  - elevation, [674](#)
  - gnssSvId, [674](#)
  - healthStatus, [674](#)
  - snr, [674](#)
  - svInfoMask, [674](#)
  - svListLen, [674](#)
  - svStatus, [674](#)
  - system, [674](#)
  - validMask, [674](#)
- SaveSMS
  - qaGobiApiSms.h, [1397](#)
- sbas\_almanac\_sv\_msk
  - GPSSStateInfo, [265](#)
- sbas\_ephemeris\_sv\_msk
  - GPSSStateInfo, [265](#)
- sbas\_health\_sv\_msk
  - GPSSStateInfo, [265](#)
- sbas\_visible\_sv\_msk
  - GPSSStateInfo, [266](#)
- sCell\_idx
  - nas\_PhyCaAggScellIndex, [435](#)
  - NASPhyCaAggScellIndex, [486](#)
  - PhyCaAggScellIndex, [590](#)
- sCell\_state
  - nas\_PhyCaAggScellIndType, [436](#)
  - nas\_PhyCaAggScellInfo, [439](#)
  - NASPhyCaAggScellIndType, [487](#)
  - NASPhyCaAggScellInfo, [488](#)
  - PhyCaAggScellIndType, [591](#)
  - PhyCaAggScellInfo, [593](#)
- screeningInd
  - connectNumInfo, [166](#)
- sduErrorRatio
  - LibPackUMTSQoS, [334](#)

- UMTSMinQoS, [838](#)
- UMTSQoS, [840](#)
- wds\_UMTSMinQoS, [1051](#)
- seDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-\_t, [973](#)
- seDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-\_t, [973](#)
- sePCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-\_t, [973](#)
- sePCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-\_t, [973](#)
- secActivate
  - fileAttributes, [219](#)
- secActivateMask
  - fileAttributes, [219](#)
- secChA
  - CDMAChannel, [141](#)
- secChB
  - CDMAChannel, [142](#)
- secDeactivate
  - fileAttributes, [219](#)
- secDeactivateMask
  - fileAttributes, [219](#)
- secIncrease
  - fileAttributes, [219](#)
- secIncreaseMask
  - fileAttributes, [219](#)
- SecProt
  - protocolSubtypeElement, [613](#)
- secRead
  - fileAttributes, [219](#)
- secReadMask
  - fileAttributes, [219](#)
- secWrite
  - fileAttributes, [219](#)
- secWriteMask
  - fileAttributes, [219](#)
- secdns
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- secdnsv6
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- second
  - nas\_timeInfo, [456](#)
  - nas\_UniversalTime, [461](#)
  - timeInfo, [784](#)
  - UniversalTime, [843](#)
- secondaryDNS
  - pack\_wds\_SetDefaultProfile\_t, [569](#)
- SecondaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- SecondaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- secondaryHA
  - unpack\_wds\_GetMobileIPProfile\_t, [959](#)
- SectorIDLen
  - NetworkStatEVDO, [508](#)
- selNetwork
  - nas\_servSystem, [446](#)
  - servSystem, [681](#)
- selected
  - BroadcastConfig, [121](#)
  - CDMABroadcastConfig, [141](#)
- selectedNetwork
  - NASServingSystemInfo, [495](#)
  - ServingSystemInfo, [680](#)
- selection
  - pack\_swima\_SLQSOMADMSendSelection\_t, [558](#)
- SendSMS
  - qaGobiApiSms.h, [1398](#)
- sendStatus
  - SMSAsyncRawSend\_s, [732](#)
- sensorData, [674](#)
  - flags, [675](#)
  - sensorDataLen, [676](#)
  - timeOfFirstSample, [676](#)
  - timeOffset, [676](#)
  - xAxis, [676](#)
  - yAxis, [676](#)
  - zAxis, [676](#)
- sensorDataLen
  - sensorData, [676](#)
- sensorDataUsage
  - qaGobiApiCbk.h, [1171](#)
- sensorDataUsage\_s, [676](#)
  - aidingIndicatorMask, [676](#)
  - usageMask, [676](#)
- serialNumbersInfo, [677](#)
  - esnSize, [677](#)
  - imeiSize, [677](#)
  - imeiSvnSize, [678](#)
  - meidSize, [678](#)
  - pESNString, [678](#)
  - pIMEIString, [678](#)
  - pImeiSvnString, [678](#)
  - pMEIDString, [678](#)
  - qaGobiApiDms.h, [1251](#)
- servSystem, [680](#)
  - csAttachState, [681](#)
  - numRadioInterfaces, [681](#)
  - psAttachState, [681](#)
  - radioInterface, [681](#)
  - regState, [681](#)
  - selNetwork, [681](#)
- ServerAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- serviceCategory
  - CDMABroadcastConfig, [141](#)
- serviceClassInformation
  - qaGobiApiVoice.h, [1485](#)
- serviceDomain
  - nas\_RejectReasonTlv, [442](#)

- serviceProviderName, 678
  - displayCondition, 678
  - spn, 678
  - spnLength, 678
- servingCellId
  - LTEInfoIntraFreq, 365
  - nas\_LTEInfoIntraFreq, 422
- ServingSystem
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- ServingSystemInfo, 678
  - csAttachState, 680
  - hdrPersonality, 680
  - psAttachState, 680
  - radiInterfaceList, 680
  - radiInterfaceNo, 680
  - registrationState, 680
  - selectedNetwork, 680
- sessionEndReason
  - \_packetSrvStatus, 62
  - slqsSessionStateInfo, 723
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 969
- SessionId
  - LOCStartReq, 358
  - pack\_loc\_Start\_t, 532
  - pack\_loc\_Stop\_t, 532
- sessionId
  - LOCStopReq, 358
  - QmiCbkLocPositionReportInd, 636
  - ssdatasession\_params, 752
  - unpack\_loc\_PositionRpt\_Ind\_t, 881
- sessionInfo, 681
  - omaDmConfig, 682
  - omaDmFota, 682
  - omaDmNotifications, 682
  - pack\_uim\_ChangePin\_t, 560
  - pack\_uim\_ReadTransparent\_t, 562
  - pack\_uim\_SetPinProtection\_t, 563
  - pack\_uim\_UnblockPin\_t, 565
  - pack\_uim\_VerifyPin\_t, 566
  - sessionInfoTlv, 683
  - sessionInfoTlvExt, 683
  - UIMAuthenticateReq, 811
  - UIMChangePinReq, 812
  - UIMGetFileAttributesReq, 816
  - UIMReadTransparentReq, 820
  - UIMRefreshCompleteReq, 822
  - UIMRefreshGetLastEventReq, 824
  - UIMRefreshOKReq, 825
  - UIMRefreshRegisterReq, 825
  - UIMSetPinProtectionReq, 827
  - UIMUnblockPinReq, 831
  - UIMVerifyPinReq, 832
- SessionInfoConfig
  - unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t, 944
- sessionInfoExt, 682
  - omaDmConfig, 682
  - omaDmFota, 682
- SessionInfoFota
  - unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t, 944
- SessionInfoNotification
  - unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t, 944
- sessionInfoTlv, 682
  - sessionInfo, 683
  - sessionType, 683
  - TlvPresent, 683
- sessionInfoTlvExt, 683
  - sessionInfo, 683
  - sessionType, 683
  - TlvPresent, 683
- sessionInformation
  - qaGobiApiCbk.h, 1171
- sessionInformationExt
  - qaGobiApiCbk.h, 1172
- SessionState
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- sessionStatus
  - omaDmNotificationsTlv, 518
  - QmiCbkLocPositionReportInd, 636
  - unpack\_loc\_PositionRpt\_Ind\_t, 881
  - unpack\_omaDmNotificationsTlv\_t, 917
- SessionType
  - pack\_swima\_SLQSOMADMGetSessionInfo\_t, 557
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- sessionType
  - omaDmFotaTlv, 516
  - pack\_swima\_SLQSOMADMCancelSession\_t, 557
  - pack\_swima\_SLQSOMADMStartSession\_t, 559
  - sessionInfoTlv, 683
  - sessionInfoTlvExt, 683
  - uim\_sessionInformation, 805
  - uim\_UIMSessionInformation, 808
  - UIMRefreshEvent, 823
  - UIMSessionInformation, 826
  - unpack\_omaDmFotaTlv\_t, 917
- set\_fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 556
  - SwiLocSetAutoStartReq, 759
- set\_fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 556
  - SwiLocSetAutoStartReq, 760
- set\_function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 556
  - SwiLocSetAutoStartReq, 760
- set\_max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 556
  - SwiLocSetAutoStartReq, 760
- set\_max\_time

- pack\_swiloc\_SwiLocSetAutoStart\_t, 556
- SwiLocSetAutoStartReq, 760
- SetACCOLC
  - qaGobiApiNas.h, 1344
- SetActivationStatusCallback
  - qaGobiApiCbk.h, 1201
- SetActiveMobileIPProfile
  - qaGobiApiWds.h, 1521
- SetAudioPathConfigReq, 683
  - pCodecSTGain, 685
  - pDTMFXTGain, 685
  - pECMode, 685
  - pNSEnable, 685
  - pRXAGCList, 685
  - pRXAVCAGCSwitch, 685
  - pRXAVCList, 685
  - pRXPCMIIRFtr, 685
  - pTXAGCList, 685
  - pTXAVCSwitch, 685
  - pTXGain, 685
  - pTXPCMIIRFtr, 685
  - Profile, 685
- SetAudioProfileReq, 685
  - EarMute, 686
  - Generator, 686
  - MicMute, 686
  - Profile, 686
  - Volume, 686
- SetAudioVolTLBConfigReq, 686
  - Generator, 687
  - Item, 687
  - Profile, 687
  - VolValue, 687
  - Volume, 687
- SetAudioVolTLBConfigResp, 687
  - ResCode, 688
- SetAutoconnect
  - qaGobiApiWds.h, 1521
- SetCATEventCallback
  - qaGobiApiCbk.h, 1201
- SetCDMANetworkParameters
  - qaGobiApiNas.h, 1345
- setCustomSettingV2, 688
  - cust\_id, 688
  - cust\_value, 688
  - value\_length, 688
- SetDataCapabilitiesCallback
  - qaGobiApiCbk.h, 1202
- SetDefaultProfile
  - qaGobiApiWds.h, 1522
- SetDefaultProfileLTE
  - qaGobiApiWds.h, 1523
- SetDefaultProfileLTEV2
  - qaGobiApiWds.h, 1525
- SetDefaultProfileNum
  - qaGobiApiWds.h, 1526
- SetDeviceStateChangeCbk
  - qaGobiApiCbk.h, 1203
- setDyingGaspCfg, 688
  - pDestSMSContent, 689
  - pDestSMSNum, 689
- SetFwDldCompletionCbk
  - qaGobiApiCbk.h, 1203
- SetGPSCallback
  - qaGobiApiCbk.h, 1204
- SetIMSSMSConfigReq, 689
  - pPhoneCtxtURI, 690
  - pPhoneCtxtURILen, 690
  - pSMSFormat, 690
  - pSMSOverIPNwInd, 690
- SetIMSSMSConfigResp, 690
  - pSettingResp, 690
- SetIMSUserConfigReq, 690
  - pIMSDomain, 690
  - pIMSDomainLen, 691
- SetIMSUserConfigResp, 691
  - pSettingResp, 691
- SetIMSVoIPConfigReq, 691
  - pAmrMode, 693
  - pAmrOctetAligned, 693
  - pAmrWBMode, 693
  - pAmrWBOctetAligned, 693
  - pAmrWbEnable, 693
  - pMinSessionExpiryTimer, 693
  - pRTPRTCPInactTimer, 693
  - pRingBackTimer, 693
  - pRingingTimer, 693
  - pScrAmrEnable, 693
  - pScrAmrWbEnable, 693
  - pSessionExpiryTimer, 693
- SetIMSVoIPConfigResp, 693
  - pSettingResp, 694
- SetImagesPreference
  - qaGobiApiFms.h, 1292
- setIndicationRegReq
  - qaGobiApiSms.h, 1395
- SetLURejectCallback
  - qaGobiApiCbk.h, 1207
- SetLocBestAvailPosCallback
  - qaGobiApiCbk.h, 1204
- SetLocCradleMountCallback
  - qaGobiApiCbk.h, 1204
- SetLocDeleteAssistDataCallback
  - qaGobiApiCbk.h, 1205
- SetLocEngineStateCallback
  - qaGobiApiCbk.h, 1205
- SetLocEventPositionCallback
  - qaGobiApiCbk.h, 1205
- SetLocEventTimeSyncCallback
  - qaGobiApiCbk.h, 1205
- SetLocGnssSvInfoCallback
  - qaGobiApiCbk.h, 1206
- SetLocInjectSensorDataCallback
  - qaGobiApiCbk.h, 1206
- SetLocInjectTimeCallback
  - qaGobiApiCbk.h, 1206

- SetLocOpModeCallback
  - qaGobiApiCbK.h, [1207](#)
- SetLocSensorStreamingCallback
  - qaGobiApiCbK.h, [1207](#)
- SetLocSetExtPowerConfigCallback
  - qaGobiApiCbK.h, [1207](#)
- SetM2MAVMuteReq, [697](#)
  - EarMute, [697](#)
  - MicMute, [698](#)
  - pCwtMute, [698](#)
  - Profile, [698](#)
- SetM2MAudioAVCFGRReq, [694](#)
  - Device, [694](#)
  - PIFACEId, [694](#)
  - pPCMPParams, [694](#)
  - Profile, [695](#)
- SetM2MAudioLPBKReq, [695](#)
  - Enable, [695](#)
- SetM2MAudioProfileReq, [695](#)
  - pCwtMute, [696](#)
  - pEarMute, [696](#)
  - pGenerator, [696](#)
  - pMicMute, [696](#)
  - pVolume, [696](#)
  - Profile, [696](#)
- SetM2MAudioVolumeReq, [696](#)
  - Generator, [697](#)
  - Level, [697](#)
  - Profile, [697](#)
- SetM2MSprkGainReq, [698](#)
  - Profile, [698](#)
  - Value, [698](#)
- SetMobileIP
  - qaGobiApiWds.h, [1527](#)
- SetMobileIPParameters
  - qaGobiApiWds.h, [1527](#)
- SetMobileIPProfile
  - qaGobiApiWds.h, [1528](#)
- SetMobileIPStatusCallback
  - qaGobiApiCbK.h, [1208](#)
- SetNMEACallback
  - qaGobiApiCbK.h, [1210](#)
- SetNasLTECphyCaIndCallback
  - qaGobiApiCbK.h, [1208](#)
- SetNetChangeCbK
  - qaGobiApiCbK.h, [1209](#)
- SetNetworkPreference
  - qaGobiApiNas.h, [1346](#)
- SetNewSMSCallback
  - qaGobiApiCbK.h, [1209](#)
- SetOMADMStateCallback
  - qaGobiApiCbK.h, [1210](#)
- SetPDSDDefaults
  - qaGobiApiPds.h, [1372](#)
- SetPDSSState
  - qaGobiApiPds.h, [1373](#)
- SetPDSSStateCallback
  - qaGobiApiCbK.h, [1210](#)
- setPINProtection, [698](#)
  - pinID, [699](#)
  - pinLength, [699](#)
  - pinOperation, [699](#)
  - pinValue, [699](#)
- SetPortAutomaticTracking
  - qaGobiApiPds.h, [1374](#)
- SetPower
  - qaGobiApiDms.h, [1263](#)
- SetPowerCallback
  - qaGobiApiCbK.h, [1211](#)
- SetRFInfoCallback
  - qaGobiApiCbK.h, [1211](#)
- SetRMTransferStatisticsCallback
  - qaGobiApiCbK.h, [1211](#)
- SetRankIndicatorCallback
  - qaGobiApiCbK.h, [1211](#)
- SetRegMgrConfigReq, [699](#)
  - pCSCFPortName, [700](#)
  - pCSCFPortNameLen, [700](#)
  - pIMSTestMode, [700](#)
  - pPriCSCFPort, [700](#)
- SetRegMgrConfigResp, [700](#)
  - pSettingResp, [700](#)
- SetRoamingIndicatorCallback
  - qaGobiApiCbK.h, [1212](#)
- SetSDKImagePath
  - qaGobiApiDcs.h, [1241](#)
- SetSIPConfigReq, [705](#)
  - pSIPLocalPort, [706](#)
  - pSigCompEnabled, [706](#)
  - pSubscribeTimer, [706](#)
  - pTimerSIPReg, [706](#)
  - pTimerT1, [706](#)
  - pTimerT2, [706](#)
  - pTimerTf, [706](#)
- SetSIPConfigResp, [706](#)
  - pSettingResp, [707](#)
- SetSLQSOMADMAAlertCallback
  - qaGobiApiCbK.h, [1213](#)
- SetSLQSOMADMAAlertCallbackExt
  - qaGobiApiCbK.h, [1213](#)
- SetSMSCAddress
  - qaGobiApiSms.h, [1398](#)
- SetSMSWake
  - qaGobiApiRms.h, [1388](#)
- SetServiceAutomaticTracking
  - qaGobiApiPds.h, [1374](#)
- SetSignalStrengthCallback
  - qaGobiApiCbK.h, [1212](#)
- setSignalStrengthInfo, [700](#)
  - pCDMAECIODelta, [704](#)
  - pCDMAECIOThresh, [704](#)
  - pCDMARSSIDelta, [704](#)
  - pCDMARSSIThresh, [704](#)
  - pGSMRSSIDelta, [704](#)
  - pGSMRSSIThresh, [704](#)
  - pHDRECIODelta, [704](#)

- pHDRECIOTresh, [704](#)
- pHDRIODelta, [704](#)
- pHDRIOTresh, [704](#)
- pHRRSSIDelta, [704](#)
- pHRRSSITresh, [704](#)
- pHRSINRDelta, [704](#)
- pHRSINRTresh, [704](#)
- pLTERSRPDelta, [704](#)
- pLTERSRPTresh, [704](#)
- pLTERSRQDelta, [704](#)
- pLTERSRQThresh, [704](#)
- pLTERSSIDelta, [705](#)
- pLTERSSITresh, [705](#)
- pLTESNRDelta, [705](#)
- pLTESNRThresh, [705](#)
- pLTESigRptConfig, [705](#)
- pTDSCDMAECIODelta, [705](#)
- pTDSCDMAECIOTresh, [705](#)
- pTDSCDMARSCPDelta, [705](#)
- pTDSCDMARSCPThresh, [705](#)
- pTDSCDMARSSIDelta, [705](#)
- pTDSCDMARSSITresh, [705](#)
- pTDSCDMASINRDelta, [705](#)
- pTDSCDMASINRTresh, [705](#)
- pWCDMAECIODelta, [705](#)
- pWCDMAECIOTresh, [705](#)
- pWCDMARSSIDelta, [705](#)
- pWCDMARSSITresh, [705](#)
- SetUSSDNoWaitIndicationCallback
  - qaGobiApiCbK.h, [1214](#)
- SetUSSDNotificationCallback
  - qaGobiApiCbK.h, [1214](#)
- SetUSSDReleaseCallback
  - qaGobiApiCbK.h, [1215](#)
- SetUimSlotStatusChangeCallback
  - qaGobiApiCbK.h, [1214](#)
- SetXTRAAutomaticDownload
  - qaGobiApiPds.h, [1375](#)
- SetXTRANetwork
  - qaGobiApiPds.h, [1375](#)
- SetupEventList
  - CatEventListTlv, [139](#)
- Severity
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [946](#)
- severity
  - omaDmFotaTlv, [516](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- Short Message Service (SMS), [36](#)
- shortName
  - nasPLMNNameResp, [491](#)
  - PLMNNetworkNameData, [597](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [895](#)
- shortNameCI
  - nasPLMNNameResp, [491](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [895](#)
- shortNameEn
  - nasPLMNNameResp, [491](#)
- unpack\_nas\_SLQSGetPLMNName\_t, [895](#)
- shortNameLen
  - nasPLMNNameResp, [491](#)
  - PLMNNetworkNameData, [597](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [895](#)
- shortNameSB
  - nasPLMNNameResp, [491](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [895](#)
- shortNameSpareBits
  - PLMNNetworkNameData, [597](#)
- sid
  - CDMAInfo, [143](#)
  - nas\_CDMAInfo, [389](#)
  - sidNid, [708](#)
  - unpack\_nas\_GetHomeNetwork\_t, [886](#)
- SidNid
  - homeSIDNID, [282](#)
- sidNid, [707](#)
  - nid, [708](#)
  - sid, [708](#)
- SigInd
  - LibPackUMTSReqQoSSigInd, [335](#)
  - UMTSReqQoSSigInd, [841](#)
- sigInfo, [708](#)
  - pECIOTresh, [709](#)
  - pHRSINRTresh, [709](#)
  - pIOTresh, [709](#)
  - pLTESNRThresh, [709](#)
  - pLTESigRptCfg, [709](#)
  - pRSRPThresh, [709](#)
  - pRSRQThresh, [709](#)
  - pRSSITresh, [709](#)
  - pTDSCDMASINRCONFTresh, [709](#)
- signal
  - signalInfo, [710](#)
- signalInfo, [709](#)
  - alertPitch, [710](#)
  - signal, [710](#)
  - signalType, [710](#)
- signalStrength
  - nas\_SignalStrengthTlv, [447](#)
- SignalStrengthDataType, [710](#)
  - thresholds, [710](#)
  - thresholdsSize, [710](#)
- signalStrengthReqMask
  - slqsSignalStrengthInfo, [725](#)
  - unpack\_nas\_SLQSGetSignalStrength\_t, [898](#)
- signalType
  - signalInfo, [710](#)
- SimCapability
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
- simCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, [847](#)
- sinr
  - HDRSSInfo, [279](#)
  - hdrSSInfo, [280](#)
  - nas\_SLQSSignalStrengthsInformation, [449](#)
  - slqsSignalStrengthInfo, [726](#)

- SLQSSignalStrengthsInformation, 729
- TDSCDMASigInfoExt, 778
- tdscdmaSigInfoExt, 779
- unpack\_nas\_SLQSGetSignalStrength\_t, 898
- sinrDelta
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- sinrThresholdList
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- sinrThresholdListLen
  - nas\_SLQSSignalStrengthsIndReq, 448
  - SLQSSignalStrengthsIndReq, 727
- sku\_str
  - slqsfwinfo\_s, 717
  - unpack\_dms\_GetFirmwareInfo\_t, 849
- slot
  - UIMPowerDownReq, 819
  - UIMPowerUpReq, 819
- slot\_t, 710
  - bICCID, 711
  - bICCIDLength, 711
  - bLogicalSlot, 711
  - uPhyCardStatus, 711
  - uPhySlotStatus, 711
- slotInf, 711
  - AppStatus, 712
  - cardState, 712
  - errorState, 712
  - numApp, 712
  - upinRetries, 712
  - upinState, 713
  - upukRetries, 713
- SlotInfo
  - cardStatus, 135
  - uim\_cardStatus, 800
- slotInfo, 713
  - AppStatus, 714
  - cardState, 714
  - errorState, 714
  - numApp, 714
  - upinRetries, 714
  - upinState, 714
  - upukRetries, 714
- slots\_t, 714
  - uimSlotStatus, 714
- slotsstatusChange
  - UIMSlotStatusChangeInfo, 829
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, 952
- slqs3GPPConfigItem
  - qaGobiApiWds.h, 1507
- SlqsNas3GppNetworkInfo, 717
  - Description, 718
  - Forbidden, 718
  - InUse, 718
  - MCC, 718
  - MNC, 718
  - Preferred, 718
  - Roaming, 718
- SlqsNas3GppNetworkRAT
  - qaGobiApiNas.h, 1322
- SlqsNasPcsDigit, 718
  - includes\_pcs\_digit, 719
  - MCC, 719
  - MNC, 719
- slqsNetworkScanInfo
  - qaGobiApiNas.h, 1322
- SlqsProfile3GPP
  - unpackWdsProfileParam, 973
  - wds\_profileInfo, 1048
  - WdsProfileParam, 1064
- SlqsProfile3GPP2
  - unpackWdsProfileParam, 973
  - wds\_profileInfo, 1048
  - WdsProfileParam, 1064
- slqsSessionStateInfo, 722
  - pQmilInterfaceInfo, 723
  - reconfiguration\_required, 723
  - sessionEndReason, 723
  - state, 723
- slqsSignalStrengthInfo, 723
  - ecioList, 725
  - ecioListLen, 725
  - errorRateList, 725
  - errorRateListLen, 725
  - Io, 725
  - ltsr, 725
  - ltsnr, 725
  - rsrqInfo, 725
  - rxSignalStrengthList, 725
  - rxSignalStrengthListLen, 725
  - signalStrengthReqMask, 725
  - sinr, 726
- slqsWdsEventInfo, 729
  - pDataBearer, 730
  - pDormancyStatus, 730
  - pPacketsCountRX, 730
  - pPacketsCountTX, 730
  - pQmilInterfaceInfo, 730
  - pTotalBytesRX, 731
  - pTotalBytesTX, 731
- slqsautoconnect, 714
  - acroamsetting, 715
  - acsetting, 715
  - action, 715
- slqsfwinfo\_s, 716
  - appversion\_str, 717
  - bootversion\_str, 717
  - carrier\_str, 717
  - cur\_carr\_name, 717
  - cur\_carr\_rev, 717
  - modelid\_str, 717
  - packageid\_str, 717
  - priversion\_str, 717
  - sku\_str, 717

- slqssendasynsmsparams\_s, 719
  - messageFormat, 721
  - messageSize, 721
  - pFollowOnDC, 721
  - pForceOnDC, 721
  - pLinktimer, 721
  - pMessage, 721
  - pRetryMessage, 721
  - pRetryMessageld, 721
  - pServiceOption, 721
  - pSmsOnlms, 721
  - pUserData, 721
- slqssendsmsparams\_s, 721
  - messageFailureCode, 722
  - messageFormat, 722
  - messageID, 722
  - messageSize, 722
  - pLinktimer, 722
  - pMessage, 722
  - pSmsOnlms, 722
- sms.h
  - LIBPACK\_QMI\_CBK\_PARAM\_NOCHANGE, 1565
  - LIBPACK\_QMI\_CBK\_PARAM\_RESET, 1565
  - LIBPACK\_QMI\_CBK\_PARAM\_SET, 1565
- sms.h, 1561
  - eqmiCbkSetStatus, 1565
  - MAX\_MSE\_TWS\_MSG, 1563
  - MAX\_SMS\_LIST\_SIZE, 1563
  - pack\_sms\_SLQSDDeleteSMS, 1566
  - pack\_sms\_SLQSGetSMS, 1566
  - pack\_sms\_SLQSGetSMSList, 1566
  - pack\_sms\_SLQSMModifySMSStatus, 1567
  - pack\_sms\_SendSMS, 1565
  - pack\_sms\_SetNewSMSCallback, 1565
  - sMSCAddressInfo, 1563
  - sMSEtwsMessageInfo, 1563
  - sMSEtwsPlmnInfo, 1564
  - sMSMTMessageInfo, 1564
  - sMSMessageModelInfo, 1564
  - sMSOnIMSInfo, 1564
  - sMSTransferRouteMTMessageInfo, 1564
  - unpack\_sms\_SLQSDDeleteSMS, 1568
  - unpack\_sms\_SLQSGetSMS, 1569
  - unpack\_sms\_SLQSGetSMSList, 1569
  - unpack\_sms\_SLQSMModifySMSStatus, 1569
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind, 1570
  - unpack\_sms\_SendSMS, 1567
  - unpack\_sms\_SetNewSMSCallback, 1568
  - unpack\_sms\_SetNewSMSCallback\_ind, 1568
- smsEventType
  - SMSEventInfo\_s, 738
- smsMaxStorageSizeReq, 738
  - pMessageMode, 738
  - storageType, 738
- smsMaxStorageSizeResp, 738
  - freeSlots, 739
  - maxStorageSize, 739
- smsMsgprotocolResp, 740
  - msgProtocol, 741
- smsOnIMS
  - SMSONIMS, 742
  - sMSOnIMS, 742
- smsRouteEntry, 743
  - messageClass, 744
  - messageType, 744
  - receiptAction, 744
  - routeStorage, 744
- smsSetRoutesReq, 744
  - numOfRoutes, 744
  - pTransferStatusReport, 744
  - routeList, 744
- snr
  - LTESSInfo, 375
  - lteSSInfo, 375
  - satelliteInfo, 674
- snrlevel
  - lteSnrinformation, 372
  - nas\_lteSnrinformation, 427
- soMask
  - DataBearerTech, 187
  - dataBearerTechnology, 189
  - qmiWSDDataBearerTechnology, 644
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- Source
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- source
  - \_getResetInfoNotification, 58
  - altitudeSrcInfo, 100
  - dmsSwiGetResetInfo, 205
  - unpack\_dms\_GetNetworkTime\_t, 852
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 857
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 858
- sourceIPMask
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- SourceLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- spc
  - pack\_nas\_SetACCOLC\_t, 533
  - pack\_wds\_SetMobileIPProfile\_t, 571
- Specific Absorption Rate (SAR), 42
- spn
  - nasPLMNNameResp, 491
  - serviceProviderName, 678
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- spnEncoding
  - nasPLMNNameResp, 491
  - unpack\_nas\_SLQSGetPLMNName\_t, 895
- spnLength
  - nasPLMNNameResp, 491
  - serviceProviderName, 678

- unpack\_nas\_SLQSGetPLMNName\_t, 895
- srcPortRangeEnd
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- srcPortRangeStart
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- srvCapability
  - detailSvcInfo, 196
  - nas\_detailSvcInfo, 401
  - nas\_sysInfoCommon, 452
  - sysInfoCommon, 775
- srvCapabilityValid
  - nas\_sysInfoCommon, 452
  - sysInfoCommon, 775
- srvDomain
  - nas\_sysInfoCommon, 452
  - sysInfoCommon, 775
- SrvDomainPref
  - NASServDomainPrefTlv, 493
- srvDomainValid
  - nas\_sysInfoCommon, 452
  - sysInfoCommon, 775
- srvOption
  - arrSvcOption, 113
- srvStatus
  - detailSvcInfo, 197
  - GSMSrvStatusInfo, 269
  - nas\_detailSvcInfo, 401
  - nas\_GSMSrvStatusInfo, 408
  - nas\_SrvStatusInfo, 450
  - SrvStatusInfo, 749
- SrvStatusInfo, 748
  - isPrefDataPath, 749
  - srvStatus, 749
- srxlev
  - cellParams, 155
  - gsmCellInfo, 267
  - nas\_cellParams, 395
  - nas\_gsmCellInfo, 406
  - nas\_umtsLTENbrCell, 460
  - nas\_wcdmaCellInfo, 462
  - umtsLTENbrCell, 835
  - wcdmaCellInfo, 1032
- ssdatasession\_params, 749
  - action, 751
  - failureReason, 751
  - failureReasonv4, 751
  - failureReasonv6, 751
  - instanceId, 751
  - ipfamily, 751
  - pAuthentication, 751
  - pPassword, 751
  - pProfileId3GPP, 751
  - pProfileId3GPP2, 751
  - pTechnology, 751
  - pUsername, 752
  - rcv4, 752
  - rcv6, 752
  - sessionId, 752
  - v4sessionId, 752
  - v6sessionId, 752
  - verbFailReason, 752
  - verbFailReasonType, 752
- stage
  - UIMRefreshEvent, 823
- StartPDSTrackingSessionExt
  - qaGobiApiPds.h, 1379
- State
  - NetworkStat1x, 506
  - NetworkStatEVDO, 508
- state
  - LocSetCradleMountReq, 356
  - omaDmConfigTlv, 512
  - omaDmConfigTlvExt, 514
  - omaDmFotaTlv, 516
  - omaDmFotaTlvExt, 518
  - QosFlowInfoState, 652
  - QosMap, 653
  - slqsSessionStateInfo, 723
  - unpack\_dms\_GetActivationState\_t, 844
  - unpack\_omaDmConfigTlv\_t, 915
  - unpack\_omaDmFotaTlv\_t, 917
  - unpack\_qos\_QosFlowInfoState\_t, 923
- statmask
  - pack\_wds\_GetPacketStatus\_t, 568
- StatsMask
  - TransferStatInd, 790
  - transferStatInd, 790
- statsMask
  - TrStatInd, 791
- StatsPeriod
  - TransferStatInd, 790
  - transferStatInd, 790
- statsPeriod
  - TrStatInd, 791
- Status
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 946
- status
  - delAssistDataStatus, 194
  - lteEARFCN, 360
  - ltePCI, 368
  - pack\_sms\_SetNewSMSCallback\_t, 552
  - QmiCbkLocBestAvailPosInd, 625
  - QmiCbkLocInjectPositionInd, 627
  - QmiCbkLocInjectUTCTimeInd, 630
  - QmiCbkLocSetExtPowerConfigInd, 637
  - unpack\_loc\_BestAvailPos\_Ind\_t, 874
  - unpack\_loc\_SetExtPowerConfig\_Ind\_t, 882
  - unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, 927
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 928
  - wcdmaUARFCN, 1042
- statusChange

- UIMStatusChangeInfo, [829](#)
- StopPDSTrackingSession
  - qaGobiApiPds.h, [1380](#)
- storageIndex
  - FMSImageIdElement, [223](#)
  - ImageIdElement, [285](#)
- storageType
  - pack\_sms\_SLQSDDeleteSMS\_t, [553](#)
  - pack\_sms\_SLQSGetSMS\_t, [553](#)
  - pack\_sms\_SLQSGetSMSList\_t, [554](#)
  - pack\_sms\_SLQSMModifySMSStatus\_t, [555](#)
  - smsMaxStorageSizeReq, [738](#)
  - SMSMemoryInfo, [739](#)
  - SMSMTMessage, [741](#)
  - smsMTMessage, [741](#)
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind-  
\_t, [941](#)
- String
  - unpack\_dms\_GetDeviceHardwareRev\_t, [847](#)
  - unpack\_dms\_GetDeviceMfr\_t, [847](#)
  - unpack\_dms\_GetFSN\_t, [850](#)
  - unpack\_dms\_UIMGetICCID\_t, [866](#)
- stringSize
  - unpack\_dms\_GetDeviceHardwareRev\_t, [847](#)
  - unpack\_dms\_GetDeviceMfr\_t, [847](#)
  - unpack\_dms\_UIMGetICCID\_t, [866](#)
- subAddr
  - calledPartySubAdd, [125](#)
- subAddrLen
  - calledPartySubAdd, [125](#)
- subAddrType
  - calledPartySubAdd, [125](#)
- subnetMask
  - IPv4Addr, [304](#)
  - unpack\_qos\_IPv4Addr\_t, [919](#)
- SubnetMaskV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- subsType
  - voiceBindSubscriptionInfo, [982](#)
- SupUSBComps
  - unpack\_dms\_GetUSBComp\_t, [854](#)
- SuppOA
  - CUGInfo, [171](#)
- SuppPrefCUG
  - CUGInfo, [171](#)
- supportedMsgLen
  - SupportedMsgList, [752](#)
- SupportedMsgList, [752](#)
  - supportedMsgLen, [752](#)
  - supportedMsgs, [752](#)
- supportedMsgs
  - SupportedMsgList, [752](#)
- svInfoMask
  - satelliteInfo, [674](#)
- svListLen
  - satelliteInfo, [674](#)
- svStatus
  - satelliteInfo, [674](#)
- svUsedforFix
  - qaGobiApiCbk.h, [1175](#)
- svUsedforFix\_s, [755](#)
  - gnssSvUsedList, [755](#)
  - gnssSvUsedList\_len, [755](#)
- svc
  - pack\_qmi\_t, [549](#)
- SvcClass
  - callFWExtInfo, [129](#)
  - callFWInfo, [130](#)
- SvcStatus
  - callFWExtInfo, [129](#)
  - callFWInfo, [130](#)
- svcType
  - ccSUPSType, [140](#)
  - SUPSType, [753](#)
- sw1
  - cardResult, [134](#)
  - uim\_cardResult, [798](#)
- sw2
  - cardResult, [134](#)
  - uim\_cardResult, [798](#)
- SwiDataTypes.h, [1570](#)
  - BOOL, [1571](#)
  - BYTE, [1571](#)
  - CHAR, [1571](#)
  - FLOAT, [1571](#)
  - INT32, [1571](#)
  - INT8, [1571](#)
  - LPCSTR, [1571](#)
  - SHORT, [1571](#)
  - SWI\_API, [1571](#)
  - ULONG, [1571](#)
  - ULONGLONG, [1571](#)
  - UNUSEDPARAM, [1571](#)
  - USHORT, [1571](#)
  - WORD, [1571](#)
- SwiLocGetAutoStart
  - qaGobiApiLoc.h, [1315](#)
- SwiLocGetAutoStartResp, [756](#)
  - fix\_rate, [758](#)
  - fix\_rate\_reported, [758](#)
  - fix\_type, [758](#)
  - fix\_type\_reported, [758](#)
  - function, [758](#)
  - function\_reported, [758](#)
  - max\_dist, [758](#)
  - max\_dist\_reported, [758](#)
  - max\_time, [758](#)
  - max\_time\_reported, [758](#)
- SwiLocSetAutoStart
  - qaGobiApiLoc.h, [1316](#)
- SwiLocSetAutoStartReq, [758](#)
  - fix\_rate, [759](#)
  - fix\_type, [759](#)
  - function, [759](#)
  - max\_dist, [759](#)
  - max\_time, [759](#)

- set\_fix\_rate, [759](#)
- set\_fix\_type, [760](#)
- set\_function, [760](#)
- set\_max\_dist, [760](#)
- set\_max\_time, [760](#)
- swiModemStatusResp, [760](#)
  - commonInfo, [760](#)
  - pLTEInfo, [760](#)
- SwiOTAMsg
  - qaGobiApiCbK.h, [1176](#)
- SwiOTAMsg\_s, [760](#)
  - data, [761](#)
  - data\_len, [761](#)
  - pLteNasRelInfo, [761](#)
  - pTime, [761](#)
  - type, [761](#)
- swiPDPRuntimeSettingsReq, [761](#)
  - contextId, [762](#)
  - contextType, [762](#)
- swiPDPRuntimeSettingsResp, [762](#)
  - pAPNName, [764](#)
  - pBearerId, [764](#)
  - pContextId, [764](#)
  - pIPv4Address, [764](#)
  - pIPv4GWAddress, [764](#)
  - pIPv6Address, [764](#)
  - pIPv6GWAddress, [764](#)
  - pPrDNSIPv4Address, [764](#)
  - pPrDNSIPv6Address, [764](#)
  - pPrPCSCFIPv4Address, [764](#)
  - pPrPCSCFIPv6Address, [764](#)
  - pSeDNSIPv4Address, [764](#)
  - pSeDNSIPv6Address, [764](#)
  - pSePCSCFIPv4Address, [764](#)
  - pSePCSCFIPv6Address, [764](#)
- swiQosFilter, [764](#)
  - index, [766](#)
  - pEspSpi, [766](#)
  - pIPv4DstAddr, [766](#)
  - pIPv4SrcAddr, [766](#)
  - pIPv6DstAddr, [766](#)
  - pIPv6Label, [766](#)
  - pIPv6SrcAddr, [766](#)
  - pIPv6TrafCls, [766](#)
  - pId, [766](#)
  - pNxtHdrProto, [766](#)
  - pPrecedence, [766](#)
  - pTCPDstPort, [766](#)
  - pTCPSrcPort, [766](#)
  - pTos, [766](#)
  - pTranDstPort, [766](#)
  - pTranSrcPort, [766](#)
  - pUDPDstPort, [766](#)
  - pUDPSrcPort, [766](#)
  - version, [767](#)
- swiQosFlow, [767](#)
  - index, [769](#)
  - p3GPP2Pri, [769](#)
  - p3GPPImCn, [769](#)
  - p3GPPResResidualBER, [769](#)
  - p3GPPSigInd, [769](#)
  - p3GPPTraHdlPri, [769](#)
  - pDataRate, [769](#)
  - pJitter, [769](#)
  - pLatency, [769](#)
  - pLteQci, [769](#)
  - pMaxAllowedPktSz, [769](#)
  - pMinPolicedPktSz, [770](#)
  - pPktErrRate, [770](#)
  - pProfileId3GPP2, [770](#)
  - pTokenBucket, [770](#)
  - pTrafficClass, [770](#)
- swiQosGranted, [770](#)
  - pRxFlow, [770](#)
  - pTxFlow, [770](#)
- swiQosIds, [770](#)
  - pIds, [770](#)
  - sz, [771](#)
- swiQosModifyReq, [771](#)
  - id, [771](#)
  - pRxFilter, [771](#)
  - pRxFlow, [771](#)
  - pTxFilter, [771](#)
  - pTxFlow, [771](#)
- swiQosReq, [771](#)
  - index, [772](#)
  - pRxFilter, [772](#)
  - pRxFlow, [772](#)
  - pTxFilter, [772](#)
  - pTxFlow, [772](#)
- swiRMTrasferStaticsReq, [772](#)
  - bResetStatistics, [773](#)
  - ulMask, [773](#)
- swiloc.h, [1571](#)
  - pack\_swiloc\_SwiLocGetAutoStart, [1572](#)
  - pack\_swiloc\_SwiLocSetAutoStart, [1572](#)
  - unpack\_swiloc\_SwiLocGetAutoStart, [1572](#)
  - unpack\_swiloc\_SwiLocSetAutoStart, [1573](#)
- swioma.h, [1573](#)
  - pack\_swioama\_SLQSOMADMAAlertCallback, [1574](#)
  - pack\_swioama\_SLQSOMADMCancelSession, [1575](#)
  - pack\_swioama\_SLQSOMADMGetSessionInfo, [1575](#)
  - pack\_swioama\_SLQSOMADMGetSettings, [1576](#)
  - pack\_swioama\_SLQSOMADMSendSelection, [1576](#)
  - pack\_swioama\_SLQSOMADMSetSettings, [1577](#)
  - pack\_swioama\_SLQSOMADMStartSession, [1578](#)
  - unpack\_swioama\_SLQSOMADMAAlertCallback, [1578](#)
  - unpack\_swioama\_SLQSOMADMAAlertCallback\_ind, [1579](#)
  - unpack\_swioama\_SLQSOMADMCancelSession, [1579](#)
  - unpack\_swioama\_SLQSOMADMGetSessionInfo, [1580](#)
  - unpack\_swioama\_SLQSOMADMGetSettings, [1580](#)

- unpack\_swioma\_SLQSOMADMSelectSendSelection, 1581
- unpack\_swioma\_SLQSOMADMSetSettings, 1581
- unpack\_swioma\_SLQSOMADMStartSession, 1581
- switchOption
  - voiceALSSetLineSwitchInfo, 981
- sysInfoCDMA
  - CDMASysInfo, 153
  - nas\_CDMASysInfo, 393
- sysInfoCommon, 773
  - isSysForbidden, 775
  - isSysForbiddenValid, 775
  - roamStatus, 775
  - roamStatusValid, 775
  - srvCapability, 775
  - srvCapabilityValid, 775
  - srvDomain, 775
  - srvDomainValid, 775
- sysInfoGSM
  - GSMSysInfo, 272
  - nas\_GSMSysInfo, 411
- sysInfoHDR
  - HDRSysInfo, 282
  - nas\_HDRSysInfo, 415
- sysInfoLTE
  - LTESysInfo, 377
  - nas\_LTESysInfo, 430
- sysInfoWCDMA
  - nas\_WCDMASysInfo, 467
  - WCDMASysInfo, 1041
- sysSelectPrefInfo
  - qaGobiApiNas.h, 1323
- sysSelectPrefParams
  - qaGobiApiNas.h, 1327
- system
  - loc\_SV, 344
  - satelliteInfo, 674
  - SV, 754
- SystemID
  - qaQmiServingSystemParam, 619
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- systemID
  - CDMASysInfo, 153
  - nas\_CDMASysInfo, 393
- systemMode
  - CommInfo, 163
  - nas\_CommInfo, 397
- sz
  - swiQosIds, 771
- t\_Sv, 776
  - entries, 776
  - len, 776
- t\_gpsTime, 775
  - gpsTimeOfWeekMs, 775
  - gpsWeek, 775
- t\_sensor, 775
  - aidingIndicatorMask, 775
  - usageMask, 775
- TCPDstPort
  - unpack\_qos\_swiQosFilter\_t, 932
- TCPsrcPort
  - unpack\_qos\_swiQosFilter\_t, 932
- TDSCDMAECIOThresh, 776
- TDSCDMAECIOThreshListLen
  - nas\_TDSCDMAECIOThresh, 453
  - TDSCDMAECIOThresh, 776
- TDSCDMARSCPTThresh, 776
- TDSCDMARSCPTThreshListLen
  - nas\_TDSCDMARSCPTThresh, 453
  - TDSCDMARSCPTThresh, 777
- TDSCDMARSSIThresh, 777
- TDSCDMARSSIThreshListLen
  - nas\_TDSCDMARSSIThresh, 454
  - TDSCDMARSSIThresh, 777
- TDSCDMASINRCONFTThresh, 779
- TDSCDMASINRThresh, 779
- TDSCDMASINRThreshListLen
  - nas\_TDSCDMASINRThresh, 455
  - TDSCDMASINRThresh, 780
- TDSCDMASigInfoExt, 777
  - ecio, 778
  - rscp, 778
  - rsi, 778
  - sinr, 778
- tFNASwiLTECphyCallInfo
  - qaGobiApiCbk.h, 1177
- tFNASwiOTAMsg
  - qaGobiApiCbk.h, 1177
- tFNActivationStatus
  - qaGobiApiCbk.h, 1176
- tFNAllCallStatus
  - qaGobiApiCbk.h, 1177
- tFNAsyncRawSend
  - qaGobiApiCbk.h, 1177
- tFNBandPreference
  - qaGobiApiCbk.h, 1178
- tFNBESTAvailPos
  - qaGobiApiCbk.h, 1179
- tFNCATEvent
  - qaGobiApiCbk.h, 1180
- tFNCbkUimSlotStatusChangeInd
  - qaGobiApiCbk.h, 1180
- tFNDHCPv4ClientLeaseStatus
  - qaGobiApiCbk.h, 1181
- tFNDTMFEvent
  - qaGobiApiCbk.h, 1182
- tFNDUNCAllInfo
  - qaGobiApiCbk.h, 1182
- tFNDataCapabilities
  - qaGobiApiCbk.h, 1180
- tFNDataSysStatus
  - qaGobiApiCbk.h, 1181
- tFNDeIAssistData
  - qaGobiApiCbk.h, 1181
- tFNDeviceStateChange

- qaGobiApiCbK.h, [1181](#)
- tFNEventPosition
  - qaGobiApiCbK.h, [1182](#)
- tFNFwDldCompletion
  - qaGobiApiCbK.h, [1182](#)
- tFNGnssSvInfo
  - qaGobiApiCbK.h, [1183](#)
- tFNHDRPersonaity
  - qaGobiApiCbK.h, [1183](#)
- tFNlmsRegMgrConfig
  - qaGobiApiCbK.h, [1184](#)
- tFNlmsSIPConfig
  - qaGobiApiCbK.h, [1184](#)
- tFNlmsSMSConfig
  - qaGobiApiCbK.h, [1184](#)
- tFNlmsUserConfig
  - qaGobiApiCbK.h, [1184](#)
- tFNlmsVoIPConfig
  - qaGobiApiCbK.h, [1185](#)
- tFNlmsaPdpStatus
  - qaGobiApiCbK.h, [1183](#)
- tFNlmsaRatStatus
  - qaGobiApiCbK.h, [1183](#)
- tFNlmsaRegStatus
  - qaGobiApiCbK.h, [1183](#)
- tFNlmsaSvcStatus
  - qaGobiApiCbK.h, [1184](#)
- tFNInfoRec
  - qaGobiApiCbK.h, [1185](#)
- tFNInjectPosition
  - qaGobiApiCbK.h, [1185](#)
- tFNInjectSensorData
  - qaGobiApiCbK.h, [1185](#)
- tFNInjectTimeStatus
  - qaGobiApiCbK.h, [1185](#)
- tFNInjectUTCTime
  - qaGobiApiCbK.h, [1185](#)
- tFNLURject
  - qaGobiApiCbK.h, [1186](#)
- tFNMemoryFull
  - qaGobiApiCbK.h, [1187](#)
- tFNMessageWaiting
  - qaGobiApiCbK.h, [1187](#)
- tFNMitiLvlRpt
  - qaGobiApiCbK.h, [1187](#)
- tFNMobileIPStatus
  - qaGobiApiCbK.h, [1187](#)
- tFNModemTempInfo
  - qaGobiApiCbK.h, [1187](#)
- tFNNet
  - qaGobiApiCbK.h, [1187](#)
- tFNNetworkTime
  - qaGobiApiCbK.h, [1188](#)
- tFNNewGPS
  - qaGobiApiCbK.h, [1188](#)
- tFNNewNMEA
  - qaGobiApiCbK.h, [1189](#)
- tFNNewRMTransferStatistics
  - qaGobiApiCbK.h, [1189](#)
- tFNNewSMS
  - qaGobiApiCbK.h, [1190](#)
- tFNOMADMState
  - qaGobiApiCbK.h, [1190](#)
- tFNOTASPStatus
  - qaGobiApiCbK.h, [1191](#)
- tFNOpMode
  - qaGobiApiCbK.h, [1190](#)
- tFNPDSSState
  - qaGobiApiCbK.h, [1191](#)
- tFNPacketSrvState
  - qaGobiApiCbK.h, [1191](#)
- tFNPower
  - qaGobiApiCbK.h, [1191](#)
- tFNPrivacyChange
  - qaGobiApiCbK.h, [1192](#)
- tFNQosNWStatus
  - qaGobiApiCbK.h, [1192](#)
- tFNQosPriEvent
  - qaGobiApiCbK.h, [1192](#)
- tFNQosStatus
  - qaGobiApiCbK.h, [1192](#)
- tFNRFIInfo
  - qaGobiApiCbK.h, [1194](#)
- tFNRankIndicator
  - qaGobiApiCbK.h, [1193](#)
- tFNResetInfo
  - qaGobiApiCbK.h, [1193](#)
- tFNRoamingIndicator
  - qaGobiApiCbK.h, [1194](#)
- tFNSDKTerminated
  - qaGobiApiCbK.h, [1194](#)
- tFNSLQSOMADMAAlert
  - qaGobiApiCbK.h, [1195](#)
- tFNSLQSQOSEvent
  - qaGobiApiCbK.h, [1196](#)
- tFNSLQSSessionState
  - qaGobiApiCbK.h, [1196](#)
- tFNSLQSSignalStrengths
  - qaGobiApiCbK.h, [1196](#)
- tFNSLQSWDSEvent
  - qaGobiApiCbK.h, [1196](#)
- tFNSMSEvents
  - qaGobiApiCbK.h, [1197](#)
- tFNSUPSInfo
  - qaGobiApiCbK.h, [1197](#)
- tFNSUPSNotification
  - qaGobiApiCbK.h, [1197](#)
- tFNSensorStreaming
  - qaGobiApiCbK.h, [1194](#)
- tFNServingSystem
  - qaGobiApiCbK.h, [1195](#)
- tFNSetCradleMount
  - qaGobiApiCbK.h, [1195](#)
- tFNSetEngineState
  - qaGobiApiCbK.h, [1195](#)
- tFNSetEventTimeSync

- qaGobiApiCbK.h, [1195](#)
- tFNSetExtPowerConfig
  - qaGobiApiCbK.h, [1195](#)
- tFNSigInfo
  - qaGobiApiCbK.h, [1195](#)
- tFNSignalStrength
  - qaGobiApiCbK.h, [1195](#)
- tFNSysInfo
  - qaGobiApiCbK.h, [1197](#)
- tFNSysSelectionPref
  - qaGobiApiCbK.h, [1197](#)
- tFNUIMRefresh
  - qaGobiApiCbK.h, [1198](#)
- tFNUIMStatusChangeInfo
  - qaGobiApiCbK.h, [1198](#)
- tFNUSSDNoWaitIndication
  - qaGobiApiCbK.h, [1199](#)
- tFNUSSDNotification
  - qaGobiApiCbK.h, [1198](#)
- tFNUSSDRelease
  - qaGobiApiCbK.h, [1199](#)
- tFNtransLayerInfo
  - qaGobiApiCbK.h, [1198](#)
- tFNtransNWRegInfo
  - qaGobiApiCbK.h, [1198](#)
- TFTIDParams, [781](#)
  - destPortRangeEnd, [782](#)
  - destPortRangeStart, [782](#)
  - eValid, [782](#)
  - filterId, [782](#)
  - flowLabel, [782](#)
  - IPSECSPi, [782](#)
  - ipVersion, [782](#)
  - nextHeader, [782](#)
  - pSourceIP, [782](#)
  - sourceIPMask, [782](#)
  - srcPortRangeEnd, [782](#)
  - srcPortRangeStart, [782](#)
  - tosMask, [782](#)
- THIRD\_INSTANCE
  - qaGobiApiCbK.h, [1165](#)
- TIME\_DATE\_BUF
  - qaGobiApiSms.h, [1393](#)
- TIME\_STAMP\_BUF
  - qaGobiApiSms.h, [1393](#)
- TPCause
  - SMSAsyncRawSend\_s, [732](#)
- TRMessageTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [939](#)
- TX\_PWR
  - NetworkStat1x, [506](#)
- TXAGCList, [792](#)
  - pTXAIG, [793](#)
  - pTXComprSlope, [793](#)
  - pTXComprThres, [793](#)
  - pTXExpSlope, [793](#)
  - pTXExpThres, [793](#)
  - pTXStaticGain, [793](#)
- TXChan
  - LTEInfo, [363](#)
  - nas\_LTEInfo, [419](#)
- tXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- TXOKBytesCount
  - DUNCallInfoInd, [209](#)
- tXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- tXOkBytesCount
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- TXPCMIIRFltr, [794](#)
  - pFlag, [795](#)
  - pStage0Val, [795](#)
  - pStage1Val, [795](#)
  - pStage2Val, [795](#)
  - pStage3Val, [795](#)
  - pStage4Val, [795](#)
  - pStageCnt, [795](#)
- tXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- tXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- tXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, [960](#)
- Tables, [53](#)
- tac
  - LTEInfoIntrafreq, [365](#)
  - LTESysInfo, [377](#)
  - nas\_LTEInfoIntrafreq, [422](#)
  - nas\_LTESysInfo, [430](#)
- tacValid
  - LTESysInfo, [377](#)
  - nas\_LTESysInfo, [430](#)
- TdsBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, [862](#)
- tdscdmaSigInfoExt, [778](#)
  - ecio, [778](#)
  - rscp, [778](#)
  - rsi, [778](#)
  - sinr, [779](#)
- tech
  - NWProfile, [511](#)
- techName
  - \_packetSrvStatus, [62](#)
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [969](#)
- techType
  - DataBearerTech, [187](#)
- Technology
  - DeviceConfigDetail, [198](#)
  - fwinf\_s, [226](#)
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- TechnologyPref
  - pack\_nas\_SetNetworkPreference\_t, [534](#)
- temperature
  - CommInfo, [163](#)
  - nas\_CommInfo, [397](#)

- temperatureData, [780](#)
- temperatureDataLen
  - temperatureData, [780](#)
- temperatureData, [780](#)
  - temperature, [780](#)
  - temperatureDataLen, [780](#)
  - timeOfFirstSample, [781](#)
  - timeOffset, [781](#)
  - timeSource, [781](#)
- textMsgLength
  - cdmaMsgEncodingParams, [148](#)
- Thermal Mitigation Device(TMD), [52](#)
- threshGsmHigh
  - lteGsmCellInfo, [361](#)
  - nas\_lteGsmCellInfo, [417](#)
- threshGsmLow
  - lteGsmCellInfo, [361](#)
  - nas\_lteGsmCellInfo, [417](#)
- threshServingLow
  - LTEInfoIntraFreq, [365](#)
  - nas\_LTEInfoIntraFreq, [422](#)
- threshXHigh
  - infoInterFreq, [303](#)
  - nas\_infoInterFreq, [416](#)
- threshXLow
  - infoInterFreq, [303](#)
  - nas\_infoInterFreq, [416](#)
- threshXhigh
  - lteWcdmaCellInfo, [378](#)
  - nas\_lteWcdmaCellInfo, [431](#)
- threshXlow
  - lteWcdmaCellInfo, [378](#)
  - nas\_lteWcdmaCellInfo, [431](#)
- thresholds
  - SignalStrengthDataType, [710](#)
- thresholdsSize
  - SignalStrengthDataType, [710](#)
- Time
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [947](#)
  - wcdmaLongMsgDecodingParams, [1035](#)
  - wcdmaMsgDecodingParams, [1036](#)
- time
  - NASTimeInfoTlv, [501](#)
- Time\_uncert\_ms
  - GPSSStateInfo, [266](#)
- timeInfo, [783](#)
  - day, [784](#)
  - dayLtSavingAdj, [784](#)
  - dayOfWeek, [784](#)
  - hour, [784](#)
  - minute, [784](#)
  - month, [784](#)
  - radiolInterface, [784](#)
  - second, [784](#)
  - timeZone, [784](#)
  - TlvPresent, [784](#)
  - year, [784](#)
- TimeLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [947](#)
- timeOfFirstSample
  - sensorData, [676](#)
  - temperatureData, [781](#)
- timeOffset
  - sensorData, [676](#)
  - temperatureData, [781](#)
- timeSource
  - temperatureData, [781](#)
- TimeStmp\_gps\_week
  - GPSSStateInfo, [266](#)
- TimeStmp\_tow\_ms
  - GPSSStateInfo, [266](#)
- timeSyncRefCounter
  - QmiCbkLocEventTimeSyncInd, [627](#)
- timeTlv
  - NASQmiCbkNasSwiOTAMessageInd, [492](#)
- timeZone
  - nas\_timeInfo, [456](#)
  - timeInfo, [784](#)
- timeout
  - pack\_qmi\_t, [549](#)
- timestamp
  - unpack\_dms\_GetNetworkTime\_t, [852](#)
- timingAdvance
  - GERANInfo, [228](#)
  - nas\_GERANInfo, [404](#)
- TlvPresent
  - CatCommonEventTlv, [137](#)
  - DataULongLongTlv, [193](#)
  - DataULongTlv, [193](#)
  - dms\_ActivationStatusTlv, [200](#)
  - dms\_OperatingModeTlv, [201](#)
  - eTWSPLMNInfoTlv, [215](#)
  - messageModeTlv, [379](#)
  - nas\_PhyCaAggPcellInfo, [434](#)
  - nas\_PhyCaAggScellIDBw, [435](#)
  - nas\_PhyCaAggScellIndex, [435](#)
  - nas\_PhyCaAggScellIndType, [436](#)
  - nas\_PhyCaAggScellInfo, [439](#)
  - nas\_RejectReasonTlv, [442](#)
  - nas\_RFInfoTlv, [443](#)
  - nas\_SignalStrengthTlv, [447](#)
  - nas\_SLQSSignalStrengthsTlv, [449](#)
  - nas\_timeInfo, [456](#)
  - NASBandPreferenceTlv, [467](#)
  - NASEmergencyModeTlv, [469](#)
  - NASGWAcqOrderPrefTlv, [477](#)
  - NASLTEBandPreferenceTlv, [481](#)
  - NASLteNasReleaseInfoTlv, [481](#)
  - NASModePreferenceTlv, [482](#)
  - NASNetSelPreferenceTlv, [482](#)
  - NASOTAMessageTlv, [484](#)
  - NASPhyCaAggPcellInfo, [485](#)
  - NASPhyCaAggScellIDBw, [486](#)
  - NASPhyCaAggScellIndex, [486](#)

- NASPhyCaAggScellIndType, [487](#)
- NASPhyCaAggScellInfo, [488](#)
- NASPRLPreferenceTlv, [492](#)
- NASRoamPreferenceTlv, [493](#)
- NASServDomainPrefTlv, [493](#)
- NASTimeInfoTlv, [501](#)
- newMTMessageTlv, [508](#)
- PhyCaAggPcellInfo, [589](#)
- PhyCaAggScellIDBw, [590](#)
- PhyCaAggScellIndex, [590](#)
- PhyCaAggScellIndType, [591](#)
- PhyCaAggScellInfo, [593](#)
- RoamingInfo, [662](#)
- sessionInfoTlv, [683](#)
- sessionInfoTlvExt, [683](#)
- sMSCAddressTlv, [734](#)
- sMSEtwsMessageTlv, [735](#)
- sMSOnIMSTlv, [743](#)
- timeInfo, [784](#)
- transferRouteMessageTlv, [789](#)
- Tlvresult
  - pack\_dms\_GetCustFeaturesV2\_t, [521](#)
  - pack\_dms\_SetCustFeaturesV2\_t, [522](#)
  - pack\_dms\_SetPower\_t, [523](#)
  - pack\_dms\_SetUSBComp\_t, [523](#)
  - pack\_dms\_UIMGetICCID\_t, [525](#)
  - pack\_fms\_GetImagesPreference\_t, [525](#)
  - pack\_fms\_GetStoredImages\_t, [525](#)
  - pack\_fms\_SetImagesPreference\_t, [526](#)
  - pack\_loc\_Delete\_Assist\_Data\_t, [527](#)
  - pack\_loc\_EventRegister\_t, [529](#)
  - pack\_loc\_SetExtPowerState\_t, [529](#)
  - pack\_loc\_SetOperationMode\_t, [530](#)
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, [530](#)
  - pack\_loc\_Start\_t, [532](#)
  - pack\_loc\_Stop\_t, [532](#)
  - pack\_nas\_SetNetworkPreference\_t, [534](#)
  - pack\_uim\_ChangePin\_t, [560](#)
  - pack\_uim\_ReadTransparent\_t, [562](#)
  - pack\_uim\_SetPinProtection\_t, [563](#)
  - pack\_uim\_UnblockPin\_t, [565](#)
  - pack\_uim\_VerifyPin\_t, [566](#)
  - unpack\_dms\_GetBandCapability\_t, [844](#)
  - unpack\_dms\_GetCrashAction\_t, [844](#)
  - unpack\_dms\_GetCustFeature\_t, [845](#)
  - unpack\_dms\_GetCustFeaturesV2\_t, [845](#)
  - unpack\_dms\_GetDeviceCap\_t, [846](#)
  - unpack\_dms\_GetDeviceHardwareRev\_t, [847](#)
  - unpack\_dms\_GetDeviceMfr\_t, [847](#)
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [848](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [849](#)
  - unpack\_dms\_GetFirmwareRevision\_t, [849](#)
  - unpack\_dms\_GetFirmwareRevisions\_t, [850](#)
  - unpack\_dms\_GetFSN\_t, [850](#)
  - unpack\_dms\_GetIMSI\_t, [851](#)
  - unpack\_dms\_GetModelID\_t, [851](#)
  - unpack\_dms\_GetNetworkTime\_t, [852](#)
  - unpack\_dms\_GetPower\_t, [852](#)
  - unpack\_dms\_GetPRLVersion\_t, [853](#)
  - unpack\_dms\_GetUSBComp\_t, [854](#)
  - unpack\_dms\_GetVoiceNumber\_t, [854](#)
  - unpack\_dms\_SetCustFeature\_t, [855](#)
  - unpack\_dms\_SetCustFeaturesV2\_t, [855](#)
  - unpack\_dms\_SetEventReport\_ind\_t, [856](#)
  - unpack\_dms\_SetEventReport\_t, [856](#)
  - unpack\_dms\_SetFirmwarePreference\_t, [856](#)
  - unpack\_dms\_SetPower\_t, [856](#)
  - unpack\_dms\_SetUSBComp\_t, [856](#)
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, [857](#)
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, [858](#)
  - unpack\_dms\_SLQSDmsSwiIndicationRegister\_t, [859](#)
  - unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t, [862](#)
  - unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, [863](#)
  - unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, [863](#)
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [865](#)
  - unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [866](#)
  - unpack\_dms\_UIMGetICCID\_t, [866](#)
  - unpack\_fms\_GetImagesPreference\_t, [867](#)
  - unpack\_fms\_GetStoredImages\_t, [868](#)
  - unpack\_fms\_SetImagesPreference\_t, [868](#)
  - unpack\_loc\_BestAvailPos\_Ind\_t, [874](#)
  - unpack\_loc\_Delete\_Assist\_Data\_t, [875](#)
  - unpack\_loc\_EngineState\_Ind\_t, [875](#)
  - unpack\_loc\_EventRegister\_t, [876](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [881](#)
  - unpack\_loc\_SetExtPowerConfig\_Ind\_t, [882](#)
  - unpack\_loc\_SetExtPowerState\_t, [883](#)
  - unpack\_loc\_SetOperationMode\_t, [883](#)
  - unpack\_loc\_SLQSLOCGetBestAvailPos\_t, [883](#)
  - unpack\_loc\_Start\_t, [884](#)
  - unpack\_loc\_Stop\_t, [884](#)
  - unpack\_nas\_GetNetworkPreference\_t, [887](#)
  - unpack\_nas\_SetNetworkPreference\_t, [892](#)
  - unpack\_nas\_SetServingSystemCallback\_ind\_t, [893](#)
  - unpack\_nas\_SlqsGetLTECphyCAInfo\_t, [894](#)
  - unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t, [910](#)
  - unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t, [910](#)
  - unpack\_uim\_ChangePin\_t, [949](#)
  - unpack\_uim\_GetCardStatus\_t, [950](#)
  - unpack\_uim\_ReadTransparent\_t, [951](#)
  - unpack\_uim\_SetPinProtection\_t, [951](#)
  - unpack\_uim\_UnblockPin\_t, [954](#)
  - unpack\_uim\_VerifyPin\_t, [955](#)
  - unpack\_wds\_SLQSCreateProfile\_t, [961](#)
  - unpack\_wds\_SLQSGetProfileSettings\_t, [965](#)
  - unpack\_wds\_SLQSSetIPFamilyPreference\_t, [968](#)
  - TmdDeRegNotMitigationLv1Req, [784](#)
  - mitigationDevID, [785](#)
  - mitigationDevIDLen, [785](#)

- TmdGetMitigationDevListResp, 785
  - pMitigationDevList, 785
  - pMitigationDevListLen, 785
- TmdGetMitigationLvlReq, 785
  - mitigationDevID, 786
  - mitigationDevIDLen, 786
- TmdGetMitigationLvlResp, 786
  - pCurrentmitigationLvl, 786
  - pReqMitigationLvl, 786
- TmdMitigationLvlIndReq, 787
  - mitigationDevID, 787
  - mitigationDevIDLen, 787
- TmdRegNotMitigationLvlReq, 787
  - mitigationDevID, 787
  - mitigationDevIDLen, 787
- toServiceId
  - BroadcastConfig, 121
- toggleMode
  - lineCtrlInfo, 336
- TokenBucket
  - unpack\_qos\_swiQosFlow\_t, 936
- tokenBucket, 788
  - bucketSz, 788
  - peakRate, 788
  - tokenRate, 788
- tokenRate
  - tokenBucket, 788
  - unpack\_qos\_tokenBucket\_t, 936
- Tos, 788
  - mask, 789
  - val, 789
- tosMask
  - LibPackTFTIDParams, 332
  - TFTIDParams, 782
- total\_rx\_bytes
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- total\_rx\_pkt
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- total\_tx\_bytes
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- total\_tx\_bytes\_drp
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- total\_tx\_pkt
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- total\_tx\_pkt\_drp
  - sQosStat, 748
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 926
- TrStatInd, 790
  - statsMask, 791
  - statsPeriod, 791
- TrackAreaCode
  - unpack\_nas\_SLQSGetServingSystem\_t, 897
- trackAreaCode
  - qaQmiServingSystemParam, 619
- TrafficClass
  - unpack\_qos\_swiQosFlow\_t, 936
- trafficClass
  - LibPackUMTSQoS, 334
  - UMTSMinQoS, 838
  - UMTSQoS, 840
  - wds\_UMTSMinQoS, 1051
- trafficPriority
  - LibPackUMTSQoS, 334
  - UMTSMinQoS, 838
  - UMTSQoS, 840
  - wds\_UMTSMinQoS, 1051
- TranDstPort
  - unpack\_qos\_swiQosFilter\_t, 932
- TranSrcPort
  - unpack\_qos\_swiQosFilter\_t, 932
- TransCap
  - \_transLayerinfo, 91
- transLayerInfo
  - qaGobiApiSms.h, 1396
- transLayerNotification
  - qaGobiApiCbk.h, 1199
- transNWRegInfoNotification
  - qaGobiApiCbk.h, 1199
- TransType
  - \_transLayerinfo, 91
- transactionID
  - SMSTransferRouteMTMessage, 746
  - sMSTransferRouteMTMessage, 745
- transferDelay
  - LibPackUMTSQoS, 334
  - UMTSMinQoS, 838
  - UMTSQoS, 840
  - wds\_UMTSMinQoS, 1051
- TransferRouteMTMessageInfo
  - transferRouteMessageTlv, 789
- transferRouteMessageTlv, 789
  - TlvPresent, 789
  - TransferRouteMTMessageInfo, 789
- TransferStatInd, 789
  - StatsMask, 790
  - StatsPeriod, 790
- transferStatInd, 790
  - StatsMask, 790
  - StatsPeriod, 790
- transferStats
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 577
- TransferStatsDataType, 790
  - interval, 790
- trueIMSI, 791
  - imsiT1112, 792
  - imsiTS1, 792
  - imsiTS2, 792
  - imsiTaddrNum, 792
  - mccT, 792
- trueSrvStatus
  - GSMSrvStatusInfo, 269

- nas\_GSMSrvStatusInfo, 408
- tx\_bytes
  - NetStats, 503
  - sQosFlowStat, 747
  - unpack\_QosFlowStat\_t, 938
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- tx\_bytes\_drp
  - sQosFlowStat, 747
  - unpack\_QosFlowStat\_t, 938
- tx\_errors
  - NetStats, 503
- tx\_overflows
  - NetStats, 503
- tx\_packets
  - NetStats, 503
- tx\_pkt
  - sQosFlowStat, 747
  - unpack\_QosFlowStat\_t, 938
- tx\_pkt\_drp
  - sQosFlowStat, 747
  - unpack\_QosFlowStat\_t, 938
- tx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- TxDropConutTlv
  - QmiCbkWdsStatisticsIndState, 640
- txInfo, 793
  - isInTraffic, 793
  - txPower, 793
- txOKBytesCount
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 965
- TxOkByteCountTlv
  - QmiCbkWdsStatisticsIndState, 640
- TxOkConutTlv
  - QmiCbkWdsStatisticsIndState, 640
- txPower
  - txInfo, 793
- TxQFilter
  - unpack\_qos\_QosFlowInfo\_t, 922
- TxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, 922
- type
  - \_getResetInfoNotification, 58
  - dmsSwiGetResetInfo, 206
  - pack\_wds\_GetDefaultProfileNum\_t, 567
  - pack\_wds\_SetDefaultProfileNum\_t, 569
  - SwiOTAMsg\_s, 761
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 857
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 858
  - unpack\_qmi\_t, 918
- u16PRLVersion
  - unpack\_dms\_GetPRLVersion\_t, 853
- u8PRLPreference
  - unpack\_dms\_GetPRLVersion\_t, 853
- UATISIZE
  - qaGobiApiNas.h, 1322
- UDPDstPort
  - unpack\_qos\_swiQosFilter\_t, 932
- UDPSrcPort
  - unpack\_qos\_swiQosFilter\_t, 932
- UIMAuthenticateReq, 810
  - authData, 811
  - pIndicationToken, 811
  - sessionInfo, 811
- UIMAuthenticateResp, 811
  - pAuthenticateResult, 811
  - pCardResult, 811
  - pIndicationToken, 811
- UIMChangePIN
  - qaGobiApiDms.h, 1276
- UIMChangePinReq, 812
  - changePIN, 812
  - pIndicationToken, 812
  - pKeyReferenceID, 812
  - sessionInfo, 812
- UIMDepersonalizationReq, 812
  - depersonalisationInfo, 813
- UIMDepersonalizationResp, 813
  - pRemainingRetries, 813
- UIMEventRegisterReqResp, 813
  - eventMask, 814
- UIMGetCardStatusResp, 814
  - pCardStatus, 814
  - pHotSwapStatus, 814
- UIMGetConfigurationReq, 814
  - pConfigurationMask, 815
- UIMGetConfigurationResp, 815
  - pAutoSelection, 816
  - pHaltSubscription, 816
  - pPersonalizationStatus, 816
- UIMGetControlKeyStatus
  - qaGobiApiDms.h, 1277
- UIMGetFileAttributesReq, 816
  - fileIndex, 816
  - pIndicationToken, 816
  - sessionInfo, 816
- UIMGetFileAttributesResp, 816
  - pCardResult, 817
  - pFileAttributes, 817
  - pIndicationToken, 817
- UIMGetICCID
  - qaGobiApiDms.h, 1278
- UIMGetPINStatus
  - qaGobiApiDms.h, 1278
- UIMGetSlotsStatusResp, 817
  - pNumberOfPhySlot, 817
  - pUimSlotsStatus, 817
- UIMPinResp, 818
  - pEncryptedPIN1, 818
  - pIndicationToken, 818
  - pRemainingRetries, 818
- UIMPowerDownReq, 818
  - slot, 819
- UIMPowerUpReq, 819

- plgnoreHotSwapSwitch, [819](#)
- slot, [819](#)
- UIMReadTransparentReq, [819](#)
  - fileIndex, [820](#)
  - pEncryptData, [820](#)
  - pIndicationToken, [820](#)
  - readTransparent, [820](#)
  - sessionInfo, [820](#)
- UIMReadTransparentResp, [820](#)
  - pCardResult, [821](#)
  - pEncryptedData, [821](#)
  - pIndicationToken, [821](#)
  - pReadResult, [821](#)
- UIMRefreshCompleteReq, [821](#)
  - refreshComplete, [822](#)
  - sessionInfo, [822](#)
- UIMRefreshEvent, [822](#)
  - aid, [823](#)
  - aidLength, [823](#)
  - arrfileInfo, [823](#)
  - mode, [823](#)
  - numOfFiles, [823](#)
  - sessionType, [823](#)
  - stage, [823](#)
- UIMRefreshGetLastEventReq, [823](#)
  - sessionInfo, [824](#)
- UIMRefreshGetLastEventResp, [824](#)
  - pRefreshEvent, [824](#)
- UIMRefreshOKReq, [824](#)
  - OKtoRefresh, [825](#)
  - sessionInfo, [825](#)
- UIMRefreshRegisterReq, [825](#)
  - regRefresh, [825](#)
  - sessionInfo, [825](#)
- UIMSessionInformation, [825](#)
  - aid, [826](#)
  - aidLength, [826](#)
  - sessionType, [826](#)
- UIMSetControlKeyProtection
  - qaGobiApiDms.h, [1279](#)
- UIMSetPINProtection
  - qaGobiApiDms.h, [1280](#)
- UIMSetPinProtectionReq, [826](#)
  - pIndicationToken, [827](#)
  - pKeyReferenceID, [827](#)
  - pinProtection, [827](#)
  - sessionInfo, [827](#)
- UIMSlotStatus, [827](#)
  - bICCID, [828](#)
  - bICCIDLength, [828](#)
  - bLogicalSlot, [828](#)
  - uPhyCardStatus, [828](#)
  - uPhySlotStatus, [828](#)
- UIMSlotStatusChangeInfo, [829](#)
  - bNumberOfPhySlots, [829](#)
  - slotsstatusChange, [829](#)
- UIMSlotsStatus, [827](#)
  - uimSlotStatus, [827](#)
- UIMStatusChangeInfo, [829](#)
  - statusChange, [829](#)
- UIMSwitchSlotReq, [829](#)
  - bLogicalSlot, [830](#)
  - uPhysicalSlot, [830](#)
- UIMUnblockControlKey
  - qaGobiApiDms.h, [1281](#)
- UIMUnblockPIN
  - qaGobiApiDms.h, [1281](#)
- UIMUnblockPinReq, [830](#)
  - pIndicationToken, [831](#)
  - pKeyReferenceID, [831](#)
  - sessionInfo, [831](#)
  - unblockPIN, [831](#)
- UIMVerifyPIN
  - qaGobiApiDms.h, [1282](#)
- UIMVerifyPinReq, [831](#)
  - pEncryptedPIN1, [832](#)
  - pIndicationToken, [832](#)
  - pKeyReferenceID, [832](#)
  - sessionInfo, [832](#)
  - verifyPIN, [832](#)
- ULONG
  - SwiDataTypes.h, [1571](#)
- ULONGLONG
  - SwiDataTypes.h, [1571](#)
- UMTSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- UMTSInfo, [832](#)
  - cellID, [833](#)
  - ecio, [833](#)
  - geranInst, [833](#)
  - GeranInstInfo, [833](#)
  - lac, [833](#)
  - plmn, [833](#)
  - psc, [834](#)
  - rsc, [834](#)
  - UMTSInstInfo, [834](#)
  - uarfcn, [834](#)
  - umtsInst, [834](#)
- UMTSInstInfo
  - nas\_UMTSInfo, [458](#)
  - UMTSInfo, [834](#)
- UMTSLTENbrCell
  - nas\_WCDMAInfoLTENeighborCell, [463](#)
  - WCDMAInfoLTENeighborCell, [1033](#)
- UMTSMinQoS, [835](#)
  - deliveryErrSDU, [837](#)
  - grntDownlinkBitrate, [837](#)
  - grntUplinkBitrate, [838](#)
  - maxDownlinkBitrate, [838](#)
  - maxSDUSize, [838](#)
  - maxUplinkBitrate, [838](#)
  - qosDeliveryOrder, [838](#)
  - resBerRatio, [838](#)
  - sduErrorRatio, [838](#)
  - trafficClass, [838](#)
  - trafficPriority, [838](#)

- transferDelay, 838
- UMTSQoS, 838
  - deliveryErrSDU, 840
  - grntDownlinkBitrate, 840
  - grntUplinkBitrate, 840
  - maxDownlinkBitrate, 840
  - maxSDUSize, 840
  - maxUplinkBitrate, 840
  - qosDeliveryOrder, 840
  - resBerRatio, 840
  - sduErrorRatio, 840
  - trafficClass, 840
  - trafficPriority, 840
  - transferDelay, 840
- UMTSReqQoS
  - LibPackUMTSReqQoSSigInd, 335
  - UMTSReqQoSSigInd, 841
- UMTSReqQoSSigInd, 840
  - SigInd, 841
  - UMTSReqQoS, 841
- UMTSInstInfo, 834
  - umtsEcio, 834
  - umtsPsc, 834
  - umtsRscp, 834
  - umtsUarfcn, 834
- UNIQUE\_ID\_LEN
  - dms.h, 1078
  - qaGobiApiDms.h, 1247
- UNUSEDPARAM
  - common.h, 1071
  - SwiDataTypes.h, 1571
- uPhyCardStatus
  - slot\_t, 711
  - UIMSlotStatus, 828
- uPhySlotStatus
  - slot\_t, 711
  - UIMSlotStatus, 828
- uResult
  - sGetDeviceSeriesResult, 707
- USBComp
  - pack\_dms\_SetUSBComp\_t, 523
  - unpack\_dms\_GetUSBComp\_t, 854
- USBCompConfig, 973
  - pUSBComp, 974
- USBCompParams, 974
  - pNumSupUSBComps, 976
  - pSupUSBComps, 976
  - pUSBComp, 976
- USHORT
  - SwiDataTypes.h, 1571
- USSD\_DCS\_8BIT
  - qaGobiApiCbk.h, 1165
- USSD\_DCS\_ASCII
  - qaGobiApiCbk.h, 1165
- USSD\_DCS\_UCS2
  - qaGobiApiCbk.h, 1165
- USSDNoWaitIndicationInfo, 976
  - pAlphaIdentifier, 977
  - pError, 977
  - pFailureCause, 977
  - pUSSDData, 977
- USSDRspFNetwork, 977
  - pRespData, 977
  - pTypeCode, 977
- USSInfo, 977
  - ussDCS, 978
  - ussData, 978
  - ussLen, 978
- USSInformation
  - voiceOrigUSSDNoWaitInfo, 1013
- USSResp, 978
  - pAlphaIDInfo, 978
  - pCCSuppsType, 978
  - pCallId, 978
  - pCcResultType, 978
  - pUSSDInfo, 979
  - pfailureCause, 978
- UUSData
  - UUSInfo, 980
- UUSDatalen
  - UUSInfo, 980
- UUSDcs
  - UUSInfo, 980
- UUSInfo, 979
  - UUSData, 980
  - UUSDatalen, 980
  - UUSDcs, 980
  - UUSType, 980
- UUSType
  - UUSInfo, 980
- uarfcn
  - lteWcdmaCellInfo, 378
  - nas\_lteWcdmaCellInfo, 431
  - nas\_UMTSInfo, 458
  - UMTSInfo, 834
  - wcdmaUARFCN, 1042
- ueInIdle
  - LTEInfoInterfreq, 363
  - LTEInfoIntrafreq, 365
  - LTEInfoNeighboringGSM, 366
  - LTEInfoNeighboringWCDMA, 367
  - nas\_LTEInfoInterfreq, 420
  - nas\_LTEInfoIntrafreq, 422
  - nas\_LTEInfoNeighboringGSM, 423
  - nas\_LTEInfoNeighboringWCDMA, 424
- uim.h, 1582
  - MAX\_ICCID\_LENGTH, 1584
  - MAX\_NO\_OF\_SLOTS, 1584
  - MAX\_SLOTS\_STATUS, 1584
  - pack\_uim\_ChangePin, 1584
  - pack\_uim\_GetCardStatus, 1584
  - pack\_uim\_ReadTransparent, 1585
  - pack\_uim\_SLQSUIEventRegister, 1585
  - pack\_uim\_SLQSUIGetSlotsStatus, 1586
  - pack\_uim\_SLQSUIMSwitchSlot, 1586
  - pack\_uim\_SetPinProtection, 1585

- pack\_uim\_UnblockPin, 1586
- pack\_uim\_VerifyPin, 1587
- unpack\_uim\_ChangePin, 1587
- unpack\_uim\_GetCardStatus, 1588
- unpack\_uim\_ReadTransparent, 1588
- unpack\_uim\_SLQSUIEventRegister, 1589
- unpack\_uim\_SLQSUIGetSlotsStatus, 1589
- unpack\_uim\_SLQSUISetStatusChangeCall-  
Back\_ind, 1590
- unpack\_uim\_SLQSUISwitchSlot, 1590
- unpack\_uim\_SetPinProtection, 1588
- unpack\_uim\_SetUimSlotStatusChangeCallback\_  
ind, 1589
- unpack\_uim\_UnblockPin, 1590
- unpack\_uim\_VerifyPin, 1591
- uim\_UIMSessionInformation, 808
  - aid, 808
  - aidLength, 808
  - sessionType, 808
- uim\_appStatus, 795
  - aidLength, 797
  - aidVal, 798
  - appState, 798
  - appType, 798
  - persoFeature, 798
  - persoRetries, 798
  - persoState, 798
  - persoUnblockRetries, 798
  - pin1Retries, 798
  - pin1State, 798
  - pin2Retries, 798
  - pin2State, 798
  - puk1Retries, 798
  - puk2Retries, 798
  - univPin, 798
- uim\_cardResult, 798
  - sw1, 798
  - sw2, 798
- uim\_cardStatus, 799
  - index1xPri, 800
  - index1xSec, 800
  - indexGwPri, 800
  - indexGwSec, 800
  - numSlot, 800
  - SlotInfo, 800
- uim\_changeUIMPIN, 800
  - oldPINLen, 800
  - oldPINVal, 801
  - pinID, 801
  - pinLen, 801
  - pinVal, 801
- uim\_encryptedPIN1, 801
  - pin1Len, 801
  - pin1Val, 801
- uim\_fileInfo, 801
  - fileID, 802
  - path, 802
  - pathLen, 802
- uim\_hotSwapStatus, 802
  - hotSwap, 802
  - hotSwapLength, 802
- uim\_readResult, 802
  - content, 803
  - contentLen, 803
- uim\_readTransparentInfo, 803
  - length, 803
  - offset, 803
- uim\_remainingRetries, 804
  - unblockLeft, 804
  - verifyLeft, 804
- uim\_sessionInformation, 804
  - aid, 805
  - aidLength, 805
  - sessionType, 805
- uim\_setPINProtection, 805
  - pinID, 806
  - pinLength, 806
  - pinOperation, 806
  - pinValue, 806
- uim\_slotInfo, 806
  - AppStatus, 807
  - cardState, 807
  - errorState, 807
  - numApp, 807
  - upinRetries, 807
  - upinState, 808
  - upukRetries, 808
- uim\_unblockUIMPIN, 808
  - newPINLen, 809
  - newPINVal, 809
  - pinID, 809
  - pukLen, 809
  - pukVal, 809
- uim\_verifyUIMPIN, 809
  - pinID, 810
  - pinLen, 810
  - pinVal, 810
- uimSlotStatus
  - slots\_t, 714
  - UIMSlotsStatus, 827
- ulData
  - DataULongTlv, 193
- ulMask
  - rmTrasferStaticsReq, 661
  - swiRMTrasferStaticsReq, 773
- ulPhysicalSlot
  - pack\_uim\_SLQSUISwitchSlot\_t, 564
  - UIMSwitchSlotReq, 830
- ulIData
  - DataULongLongTlv, 193
- umtsEcio
  - nas\_UMTSInstInfo, 459
  - UMTSInstInfo, 834
- umtsInst
  - nas\_UMTSInfo, 458
  - UMTSInfo, 834

- umtsLTENbrCell, [834](#)
  - cellsTDD, [835](#)
  - earfcn, [835](#)
  - pci, [835](#)
  - rsrp, [835](#)
  - rsrq, [835](#)
  - srxlev, [835](#)
- umtsLTENbrCellLen
  - nas\_WCDMAInfoLTENeighborCell, [463](#)
  - WCDMAInfoLTENeighborCell, [1033](#)
- umtsPsc
  - nas\_UMTSinstInfo, [459](#)
  - UMTSinstInfo, [834](#)
- umtsRscp
  - nas\_UMTSinstInfo, [459](#)
  - UMTSinstInfo, [834](#)
- umtsUarfcn
  - nas\_UMTSinstInfo, [459](#)
  - UMTSinstInfo, [834](#)
- UnPackGetProfileSettingOut, [973](#)
  - curProfile, [973](#)
  - pExtErrCode, [973](#)
- unblockLeft
  - personalizationStatus, [588](#)
  - remainingRetries, [657](#)
  - uim\_remainingRetries, [804](#)
- unblockPIN
  - UIMUnblockPinReq, [831](#)
- unblockUIMPIN, [841](#)
  - newPINLen, [842](#)
  - newPINVal, [842](#)
  - pinID, [842](#)
  - pukLen, [842](#)
  - pukVal, [842](#)
- uniqueID
  - CurrImageInfo, [176](#)
  - image\_info\_t, [283](#)
- univPin
  - appStats, [103](#)
  - appStatus, [106](#)
  - uim\_appStatus, [798](#)
- UniversalTime, [842](#)
  - day, [843](#)
  - dayOfWeek, [843](#)
  - hour, [843](#)
  - minute, [843](#)
  - month, [843](#)
  - second, [843](#)
  - year, [843](#)
- universalTime
  - nasNetworkTime, [483](#)
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind-  
\_t, [908](#)
- unpack\_QosFlowStat\_t, [937](#)
  - bearerId, [938](#)
  - tx\_bytes, [938](#)
  - tx\_bytes\_drp, [938](#)
  - tx\_pkt, [938](#)
  - tx\_pkt\_drp, [938](#)
- unpack\_dms\_GetActivationState
  - dms.h, [1093](#)
- unpack\_dms\_GetActivationState\_t, [843](#)
  - state, [844](#)
- unpack\_dms\_GetBandCapability
  - dms.h, [1093](#)
- unpack\_dms\_GetBandCapability\_t, [844](#)
  - BandCapability, [844](#)
  - Tlvresult, [844](#)
- unpack\_dms\_GetCrashAction
  - dms.h, [1093](#)
- unpack\_dms\_GetCrashAction\_t, [844](#)
  - DevCrashState, [844](#)
  - Tlvresult, [844](#)
- unpack\_dms\_GetCustFeature
  - dms.h, [1094](#)
- unpack\_dms\_GetCustFeature\_t, [844](#)
  - DHCPRelayEnabled, [845](#)
  - DisableIMSI, [845](#)
  - GPSPMP, [845](#)
  - GPSSel, [845](#)
  - GpsEnable, [845](#)
  - IPFamSupport, [845](#)
  - IsVoiceEnabled, [845](#)
  - RMAutoConnect, [845](#)
  - SMSSupport, [845](#)
  - Tlvresult, [845](#)
- unpack\_dms\_GetCustFeaturesV2
  - dms.h, [1094](#)
- unpack\_dms\_GetCustFeaturesV2\_t, [845](#)
  - GetCustomFeatureV2, [845](#)
  - Tlvresult, [845](#)
- unpack\_dms\_GetDeviceCap
  - dms.h, [1094](#)
- unpack\_dms\_GetDeviceCap\_t, [846](#)
  - DataServiceCapability, [846](#)
  - MaxRXChannelRate, [846](#)
  - MaxTXChannelRate, [846](#)
  - Radiofaces, [846](#)
  - RadiofacesSize, [846](#)
  - SimCapability, [846](#)
  - Tlvresult, [846](#)
- unpack\_dms\_GetDeviceCapabilities
  - dms.h, [1095](#)
- unpack\_dms\_GetDeviceCapabilities\_t, [846](#)
  - dataServiceCaCapability, [847](#)
  - maxRxChannelRate, [847](#)
  - maxTxChannelRate, [847](#)
  - Radiofaces, [847](#)
  - radiofacesSize, [847](#)
  - simCapability, [847](#)
- unpack\_dms\_GetDeviceHardwareRev
  - dms.h, [1095](#)
- unpack\_dms\_GetDeviceHardwareRev\_t, [847](#)
  - String, [847](#)
  - stringSize, [847](#)
  - Tlvresult, [847](#)

- unpack\_dms\_GetDeviceMfr
  - dms.h, [1095](#)
- unpack\_dms\_GetDeviceMfr\_t, [847](#)
  - String, [847](#)
  - stringSize, [847](#)
  - Tlvresult, [847](#)
- unpack\_dms\_GetDeviceSerialNumbers
  - dms.h, [1096](#)
- unpack\_dms\_GetDeviceSerialNumbers\_t, [847](#)
  - ESNString, [848](#)
  - esnSize, [848](#)
  - IMEIString, [848](#)
  - imeiSize, [848](#)
  - imeiSvnSize, [848](#)
  - ImeiSvnString, [848](#)
  - MEIDString, [848](#)
  - meidSize, [848](#)
  - Tlvresult, [848](#)
- unpack\_dms\_GetFSN
  - dms.h, [1097](#)
- unpack\_dms\_GetFSN\_t, [850](#)
  - String, [850](#)
  - Tlvresult, [850](#)
- unpack\_dms\_GetFirmwareInfo
  - dms.h, [1096](#)
- unpack\_dms\_GetFirmwareInfo\_t, [848](#)
  - appversion\_str, [849](#)
  - bootversion\_str, [849](#)
  - carrier\_str, [849](#)
  - cur\_carr\_name, [849](#)
  - cur\_carr\_rev, [849](#)
  - modelid\_str, [849](#)
  - packageid\_str, [849](#)
  - priversion\_str, [849](#)
  - sku\_str, [849](#)
  - Tlvresult, [849](#)
- unpack\_dms\_GetFirmwareRevision
  - dms.h, [1096](#)
- unpack\_dms\_GetFirmwareRevision\_t, [849](#)
  - AMSSString, [849](#)
  - amssSize, [849](#)
  - PRIStr, [849](#)
  - Tlvresult, [849](#)
- unpack\_dms\_GetFirmwareRevisions
  - dms.h, [1097](#)
- unpack\_dms\_GetFirmwareRevisions\_t, [849](#)
  - AMSSString, [850](#)
  - amssSize, [850](#)
  - bootSize, [850](#)
  - BootString, [850](#)
  - PRIStr, [850](#)
  - priSize, [850](#)
  - Tlvresult, [850](#)
- unpack\_dms\_GetHardwareRevision
  - dms.h, [1097](#)
- unpack\_dms\_GetHardwareRevision\_t, [850](#)
  - hwVer, [851](#)
- unpack\_dms\_GetIMSI
  - dms.h, [1098](#)
- unpack\_dms\_GetIMSI\_t, [851](#)
  - imsi, [851](#)
  - Tlvresult, [851](#)
- unpack\_dms\_GetModelID
  - dms.h, [1098](#)
- unpack\_dms\_GetModelID\_t, [851](#)
  - modelid, [851](#)
  - Tlvresult, [851](#)
- unpack\_dms\_GetNetworkTime
  - dms.h, [1098](#)
- unpack\_dms\_GetNetworkTime\_t, [851](#)
  - source, [852](#)
  - timestamp, [852](#)
  - Tlvresult, [852](#)
- unpack\_dms\_GetPRLVersion
  - dms.h, [1099](#)
- unpack\_dms\_GetPRLVersion\_t, [853](#)
  - Tlvresult, [853](#)
  - u16PRLVersion, [853](#)
  - u8PRLPreference, [853](#)
- unpack\_dms\_GetPower
  - dms.h, [1099](#)
- unpack\_dms\_GetPower\_t, [852](#)
  - HardwareControlledMode, [852](#)
  - OfflineReason, [852](#)
  - OperationMode, [852](#)
  - Tlvresult, [852](#)
- unpack\_dms\_GetSerialNumbers
  - dms.h, [1099](#)
- unpack\_dms\_GetSerialNumbers\_t, [853](#)
  - esn, [853](#)
  - imei\_no, [853](#)
  - imeisv\_svn, [853](#)
  - meid, [853](#)
- unpack\_dms\_GetUSBComp
  - dms.h, [1100](#)
- unpack\_dms\_GetUSBComp\_t, [853](#)
  - NumSupUSBComps, [854](#)
  - SupUSBComps, [854](#)
  - Tlvresult, [854](#)
  - USBComp, [854](#)
- unpack\_dms\_GetVoiceNumber
  - dms.h, [1100](#)
- unpack\_dms\_GetVoiceNumber\_t, [854](#)
  - MIN, [854](#)
  - minSize, [854](#)
  - Tlvresult, [854](#)
  - VoiceNumber, [854](#)
  - voiceNumberSize, [854](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo
  - dms.h, [1103](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind
  - dms.h, [1103](#)
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, [856](#)
  - source, [857](#)
  - Tlvresult, [857](#)
  - type, [857](#)

- unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 857
  - source, 858
  - Tlvresult, 858
  - type, 858
- unpack\_dms\_SLQSDmsSwiIndicationRegister
  - dms.h, 1104
- unpack\_dms\_SLQSDmsSwiIndicationRegister\_t, 858
  - Tlvresult, 859
- unpack\_dms\_SLQSSwiGetBandCapability
  - dms.h, 1104
- unpack\_dms\_SLQSSwiGetBandCapability\_t, 859
  - bandCapability, 862
  - LteBandCapability, 862
  - TdsBandCapability, 862
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics
  - dms.h, 1105
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t, 862
  - Tlvresult, 862
- unpack\_dms\_SLQSSwiGetDyingGaspCfg
  - dms.h, 1105
- unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, 862
  - pGetDyingGaspCfg, 863
  - Tlvresult, 863
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics
  - dms.h, 1105
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, 863
  - pGetDyingGaspStatistics, 863
  - Tlvresult, 863
- unpack\_dms\_SLQSSwiGetFirmwareCurr
  - dms.h, 1106
- unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 863
  - carrier, 864
  - fwvers, 864
  - numEntries, 864
  - pCurrImgInfo, 864
  - pkgver, 864
  - priver, 864
- unpack\_dms\_SLQSSwiGetFwUpdateStatus
  - dms.h, 1106
- unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 864
  - imgType, 865
  - logString, 865
  - refData, 865
  - refString, 865
  - ResCode, 865
  - Tlvresult, 865
- unpack\_dms\_SLQSSwiSetDyingGaspCfg
  - dms.h, 1106
- unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 866
  - Tlvresult, 866
- unpack\_dms\_SetCrashAction
  - dms.h, 1100
- unpack\_dms\_SetCrashAction\_t, 854
  - notused, 854
- unpack\_dms\_SetCustFeature
  - dms.h, 1101
- unpack\_dms\_SetCustFeature\_t, 854
  - Tlvresult, 855
- unpack\_dms\_SetCustFeaturesV2
  - dms.h, 1101
- unpack\_dms\_SetCustFeaturesV2\_t, 855
  - Tlvresult, 855
- unpack\_dms\_SetEventReport
  - dms.h, 1101
- unpack\_dms\_SetEventReport\_ind
  - dms.h, 1102
- unpack\_dms\_SetEventReport\_ind\_t, 855
  - ActivationStatusTlv, 856
  - OperatingModeTlv, 856
  - Tlvresult, 856
- unpack\_dms\_SetEventReport\_t, 856
  - Tlvresult, 856
- unpack\_dms\_SetFirmwarePreference
  - dms.h, 1102
- unpack\_dms\_SetFirmwarePreference\_t, 856
  - Tlvresult, 856
- unpack\_dms\_SetPower
  - dms.h, 1102
- unpack\_dms\_SetPower\_t, 856
  - Tlvresult, 856
- unpack\_dms\_SetUSBComp
  - dms.h, 1103
- unpack\_dms\_SetUSBComp\_t, 856
  - Tlvresult, 856
- unpack\_dms\_UIMGetICCID
  - dms.h, 1107
- unpack\_dms\_UIMGetICCID\_t, 866
  - String, 866
  - stringSize, 866
  - Tlvresult, 866
- unpack\_fms\_GetImagesPreference
  - fms.h, 1109
- unpack\_fms\_GetImagesPreference\_t, 867
  - ImageListSize, 867
  - pImageList, 867
  - Tlvresult, 867
- unpack\_fms\_GetStoredImages
  - fms.h, 1110
- unpack\_fms\_GetStoredImages\_t, 867
  - imageList, 868
  - imagelistSize, 868
  - Tlvresult, 868
- unpack\_fms\_SetImagesPreference
  - fms.h, 1110
- unpack\_fms\_SetImagesPreference\_t, 868
  - ImageTypes, 868
  - ImageTypesSize, 868
  - Tlvresult, 868
- unpack\_loc\_BestAvailPos\_Ind
  - loc.h, 1117
- unpack\_loc\_BestAvailPos\_Ind\_t, 868
  - pAltitudeWrtEllipsoid, 873
  - pAltitudeWrtMeanSeaLevel, 873
  - pGpsTime, 873
  - pHeading, 873

- pHeadingUnc, [873](#)
- pHorCirConf, [873](#)
- pHorEllpConf, [873](#)
- pHorReliability, [873](#)
- pHorUncCircular, [873](#)
- pHorUncEllipseOrientAzimuth, [873](#)
- pHorUncEllipseSemiMajor, [873](#)
- pHorUncEllipseSemiMinor, [873](#)
- pLatitude, [874](#)
- pLongitude, [874](#)
- pMagneticDeviation, [874](#)
- pPrecisionDilution, [874](#)
- pSensorDataUsage, [874](#)
- pSpeedHorizontal, [874](#)
- pSpeedUnc, [874](#)
- pSpeedVertical, [874](#)
- pSpeedVerticalUnc, [874](#)
- pSvUsedforFix, [874](#)
- pTechnologyMask, [874](#)
- pTimeSrc, [874](#)
- pTimeUnc, [874](#)
- pTimestampUtc, [874](#)
- pVertConfidence, [874](#)
- pVertReliability, [874](#)
- pVertUnc, [874](#)
- pXid, [874](#)
- status, [874](#)
- Tlvresult, [874](#)
- unpack\_loc\_Delete\_Assist\_Data\_t, [874](#)
  - Tlvresult, [875](#)
- unpack\_loc\_DeleteAssistData
  - loc.h, [1117](#)
- unpack\_loc\_EngineState\_Ind
  - loc.h, [1118](#)
- unpack\_loc\_EngineState\_Ind\_t, [875](#)
  - engineState, [875](#)
  - Tlvresult, [875](#)
- unpack\_loc\_EventRegister
  - loc.h, [1118](#)
- unpack\_loc\_EventRegister\_t, [875](#)
  - Tlvresult, [876](#)
- unpack\_loc\_PositionRpt\_Ind
  - loc.h, [1118](#)
- unpack\_loc\_PositionRpt\_Ind\_t, [876](#)
  - pAltitudeAssumed, [880](#)
  - pAltitudeWrtEllipsoid, [880](#)
  - pAltitudeWrtMeanSeaLevel, [880](#)
  - pFixId, [880](#)
  - pGpsTime, [880](#)
  - pHeading, [880](#)
  - pHeadingUnc, [881](#)
  - pHorConfidence, [881](#)
  - pHorReliability, [881](#)
  - pHorUncCircular, [881](#)
  - pHorUncEllipseOrientAzimuth, [881](#)
  - pHorUncEllipseSemiMajor, [881](#)
  - pHorUncEllipseSemiMinor, [881](#)
  - pLatitude, [881](#)
  - pLeapSeconds, [881](#)
  - pLongitude, [881](#)
  - pMagneticDeviation, [881](#)
  - pPrecisionDilution, [881](#)
  - pSensorDataUsage, [881](#)
  - pSpeedHorizontal, [881](#)
  - pSpeedUnc, [881](#)
  - pSpeedVertical, [881](#)
  - pSvUsedforFix, [881](#)
  - pTechnologyMask, [881](#)
  - pTimeSrc, [881](#)
  - pTimeUnc, [881](#)
  - pTimestampUtc, [881](#)
  - pVertConfidence, [881](#)
  - pVertReliability, [881](#)
  - pVertUnc, [881](#)
  - sessionId, [881](#)
  - sessionStatus, [881](#)
  - Tlvresult, [881](#)
- unpack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, [1120](#)
- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, [883](#)
  - Tlvresult, [883](#)
- unpack\_loc\_SetExtPowerConfig\_Ind
  - loc.h, [1119](#)
- unpack\_loc\_SetExtPowerConfig\_Ind\_t, [882](#)
  - status, [882](#)
  - Tlvresult, [882](#)
- unpack\_loc\_SetExtPowerState
  - loc.h, [1119](#)
- unpack\_loc\_SetExtPowerState\_t, [882](#)
  - Tlvresult, [883](#)
- unpack\_loc\_SetOperationMode
  - loc.h, [1119](#)
- unpack\_loc\_SetOperationMode\_t, [883](#)
  - Tlvresult, [883](#)
- unpack\_loc\_Start
  - loc.h, [1120](#)
- unpack\_loc\_Start\_t, [883](#)
  - Tlvresult, [884](#)
- unpack\_loc\_Stop
  - loc.h, [1120](#)
- unpack\_loc\_Stop\_t, [884](#)
  - Tlvresult, [884](#)
- unpack\_nas\_GetACCOLC
  - nas.h, [1137](#)
- unpack\_nas\_GetANAAAuthenticationStatus
  - nas.h, [1137](#)
- unpack\_nas\_GetCDMANetworkParameters
  - nas.h, [1138](#)
- unpack\_nas\_GetCDMANetworkParameters\_t, [884](#)
  - Application, [885](#)
  - Broadcast, [885](#)
  - CustomSCP, [885](#)
  - ForceRev0, [885](#)
  - Protocol, [885](#)
  - RegForeignNID, [885](#)
  - RegForeignSID, [885](#)

- RegHomeSID, [885](#)
- Roaming, [885](#)
- SCI, [885](#)
- SCM, [885](#)
- unpack\_nas\_GetHomeNetwork
  - nas.h, [1138](#)
- unpack\_nas\_GetHomeNetwork\_t, [885](#)
  - mcc, [886](#)
  - mnc, [886](#)
  - name, [886](#)
  - nid, [886](#)
  - sid, [886](#)
- unpack\_nas\_GetNetworkPreference
  - nas.h, [1138](#)
- unpack\_nas\_GetNetworkPreference\_t, [886](#)
  - ActiveTechPref, [887](#)
  - Duration, [887](#)
  - PersistentTechPref, [887](#)
  - Tlvresult, [887](#)
- unpack\_nas\_GetRFInfo
  - nas.h, [1138](#)
- unpack\_nas\_GetRFInfo\_t, [887](#)
  - instancesSize, [887](#)
  - RFBandInfoElements, [887](#)
- unpack\_nas\_GetServingNetwork
  - nas.h, [1139](#)
- unpack\_nas\_GetServingNetwork\_t, [887](#)
  - CSDomain, [888](#)
  - DataCaps, [888](#)
  - DataCapsLen, [888](#)
  - MCC, [888](#)
  - MNC, [888](#)
  - Name, [888](#)
  - nameSize, [888](#)
  - PSDomain, [888](#)
  - RAN, [888](#)
  - Radiolfaces, [888](#)
  - RadiolfacesSize, [888](#)
  - RegistrationState, [888](#)
  - Roaming, [888](#)
- unpack\_nas\_GetServingNetworkCapabilities
  - nas.h, [1139](#)
- unpack\_nas\_GetServingNetworkCapabilities\_t, [888](#)
  - DataCaps, [889](#)
  - DataCapsLen, [889](#)
- unpack\_nas\_GetSignalStrengths
  - nas.h, [1139](#)
- unpack\_nas\_GetSignalStrengths\_t, [889](#)
  - len, [889](#)
  - radio, [889](#)
  - rsi, [889](#)
- unpack\_nas\_PerformNetworkScan
  - nas.h, [1140](#)
- unpack\_nas\_PerformNetworkScan\_t, [889](#)
  - p3GppNetworkInfoInstances, [890](#)
  - p3GppNetworkInstanceSize, [890](#)
  - pPCInstance, [890](#)
  - pPCInstanceSize, [890](#)
- pRATInstance, [890](#)
- pRATInstanceSize, [890](#)
- pScanResult, [890](#)
- unpack\_nas\_SLQSGetNetworkTime
  - nas.h, [1142](#)
- unpack\_nas\_SLQSGetNetworkTime\_t, [894](#)
  - p3GPP2TimeInfo, [894](#)
  - p3GPPTIMEInfo, [894](#)
- unpack\_nas\_SLQSGetPLMNName
  - nas.h, [1142](#)
- unpack\_nas\_SLQSGetPLMNName\_t, [894](#)
  - longName, [895](#)
  - longNameCI, [895](#)
  - longNameEn, [895](#)
  - longNameLen, [895](#)
  - longNameSB, [895](#)
  - shortName, [895](#)
  - shortNameCI, [895](#)
  - shortNameEn, [895](#)
  - shortNameLen, [895](#)
  - shortNameSB, [895](#)
  - spn, [895](#)
  - spnEncoding, [895](#)
  - spnLength, [895](#)
- unpack\_nas\_SLQSGetServingSystem
  - nas.h, [1143](#)
- unpack\_nas\_SLQSGetServingSystem\_t, [895](#)
  - BasestationID, [896](#)
  - BasestationLatitude, [897](#)
  - BasestationLongitude, [897](#)
  - CDMASystemInfoExt, [897](#)
  - CallBarStatus, [897](#)
  - CellID, [897](#)
  - ConcSvcInfo, [897](#)
  - CurrentPLMN, [897](#)
  - DTMInd, [897](#)
  - DataSrvCapabilities, [897](#)
  - DefaultRoamInd, [897](#)
  - DetailedSvcInfo, [897](#)
  - Gpp2TimeZone, [897](#)
  - GppNetworkDSTAdjustment, [897](#)
  - GppTimeZone, [897](#)
  - HdrPersonality, [897](#)
  - Lac, [897](#)
  - NetworkID, [897](#)
  - PRLInd, [897](#)
  - RoamIndicatorVal, [897](#)
  - RoamingIndicatorList, [897](#)
  - ServingSystem, [897](#)
  - SystemID, [897](#)
  - TrackAreaCode, [897](#)
- unpack\_nas\_SLQSGetSignalStrength
  - nas.h, [1143](#)
- unpack\_nas\_SLQSGetSignalStrength\_t, [897](#)
  - ecioList, [898](#)
  - ecioListLen, [898](#)
  - errorRateList, [898](#)
  - errorRateListLen, [898](#)

- lo, [898](#)
- ltersrp, [898](#)
- ltesnr, [898](#)
- rsrqInfo, [898](#)
- rxSignalStrengthList, [898](#)
- rxSignalStrengthListLen, [898](#)
- signalStrengthReqMask, [898](#)
- sinr, [898](#)
- unpack\_nas\_SLQSGetSysInfo
  - nas.h, [1143](#)
- unpack\_nas\_SLQSGetSysInfo\_t, [899](#)
  - pAddCDMASysInfo, [901](#)
  - pAddGSMSysInfo, [901](#)
  - pAddHDRSysInfo, [901](#)
  - pAddLTESysInfo, [901](#)
  - pCDMASrvStatusInfo, [901](#)
  - pCDMASysInfo, [901](#)
  - pGSMCallBarringSysInfo, [901](#)
  - pGSMCIPHERDomainSysInfo, [901](#)
  - pGSMSrvStatusInfo, [901](#)
  - pGSMSysInfo, [901](#)
  - pHRSrvStatusInfo, [901](#)
  - pHRSysInfo, [901](#)
  - pLTESrvStatusInfo, [901](#)
  - pLTESysInfo, [901](#)
  - pLTEVoiceSupportSysInfo, [901](#)
  - pWCDMASysInfo, [901](#)
- unpack\_nas\_SLQSGetSysSelectionPref
  - nas.h, [1144](#)
- unpack\_nas\_SLQSGetSysSelectionPref\_t, [901](#)
  - pBandPref, [905](#)
  - pEmerMode, [905](#)
  - pGWAcqOrderPref, [905](#)
  - pLTEBandPref, [905](#)
  - pModePref, [905](#)
  - pNetSelPref, [905](#)
  - pPRLPref, [905](#)
  - pRoamPref, [905](#)
  - pSrvDomainPref, [905](#)
- unpack\_nas\_SLQSIInitiateNetworkRegistration
  - nas.h, [1144](#)
- unpack\_nas\_SLQSNasConfigSigInfo2
  - nas.h, [1144](#)
- unpack\_nas\_SLQSNasGetCellLocationInfo
  - nas.h, [1144](#)
- unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [905](#)
  - pCDMAInfo, [906](#)
  - pLTEInfoInterfreq, [906](#)
  - pLTEInfoIntrafreq, [906](#)
  - pUMTSInfo, [907](#)
- unpack\_nas\_SLQSNasGetSigInfo
  - nas.h, [1145](#)
- unpack\_nas\_SLQSNasGetSigInfo\_t, [907](#)
  - CDMASSInfo, [907](#)
  - GSMSSInfo, [907](#)
  - HDRSSInfo, [907](#)
  - LTESInfo, [907](#)
- unpack\_nas\_SLQSNasIndicationRegisterExt
  - nas.h, [1145](#)
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind
  - nas.h, [1145](#)
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t, [907](#)
  - pDayltSavAdj, [908](#)
  - pRadioInterface, [908](#)
  - pTimeZone, [908](#)
  - universalTime, [908](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind
  - nas.h, [1146](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [908](#)
  - pRscp, [909](#)
- unpack\_nas\_SLQSNasSmiModemStatus
  - nas.h, [1146](#)
- unpack\_nas\_SLQSNasSmiModemStatus\_t, [909](#)
  - commonInfo, [909](#)
  - pLTEInfo, [909](#)
- unpack\_nas\_SLQSNasSmiOTAMessageCallback
  - nas.h, [1146](#)
- unpack\_nas\_SLQSNasSmiOTAMessageCallback\_ind
  - nas.h, [1147](#)
- unpack\_nas\_SLQSNasSmiOTAMessageCallback\_ind\_t, [910](#)
  - Info, [910](#)
  - Tlvresult, [910](#)
- unpack\_nas\_SLQSNasSysInfoCallback\_ind
  - nas.h, [1147](#)
- unpack\_nas\_SLQSSetBandPreference
  - nas.h, [1147](#)
- unpack\_nas\_SLQSSetSignalStrengthsCallback
  - nas.h, [1147](#)
- unpack\_nas\_SLQSSetSysSelectionPref
  - nas.h, [1148](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind
  - nas.h, [1148](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t, [910](#)
  - Info, [910](#)
  - Tlvresult, [910](#)
- unpack\_nas\_SLQSSwiGetLteCQI
  - nas.h, [1148](#)
- unpack\_nas\_SLQSSwiGetLteCQI\_t, [910](#)
  - ValidityCW0, [911](#)
  - ValidityCW1, [911](#)
- unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [911](#)
  - pGSMSysInfo, [914](#)
  - pHRSysInfo, [914](#)
  - pLTESysInfo, [914](#)
  - pSysInfoNoChange, [914](#)
- unpack\_nas\_SetACCOLC
  - nas.h, [1140](#)
- unpack\_nas\_SetDataCapabilitiesCallback\_ind
  - nas.h, [1140](#)
- unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, [890](#)
  - dataCaps, [890](#)
  - dataCapsSize, [890](#)
- unpack\_nas\_SetEventReportInd

- nas.h, [1141](#)
- unpack\_nas\_SetEventReportInd\_t, [890](#)
  - RFTlv, [891](#)
  - RRTlv, [891](#)
  - SLQSSSTlv, [891](#)
  - SSTlv, [891](#)
- unpack\_nas\_SetLURejectCallback
  - nas.h, [1141](#)
- unpack\_nas\_SetNasLTECphyCaIndCallback\_ind
  - nas.h, [1141](#)
- unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [891](#)
- unpack\_nas\_SetNetworkPreference
  - nas.h, [1141](#)
- unpack\_nas\_SetNetworkPreference\_t, [892](#)
  - Tlvresult, [892](#)
- unpack\_nas\_SetRFInfoCallback
  - nas.h, [1141](#)
- unpack\_nas\_SetRoamingIndicatorCallback\_ind
  - nas.h, [1142](#)
- unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, [893](#)
  - roaming, [893](#)
- unpack\_nas\_SetServingSystemCallback\_ind
  - nas.h, [1142](#)
- unpack\_nas\_SetServingSystemCallback\_ind\_t, [893](#)
  - SSInfo, [893](#)
  - Tlvresult, [893](#)
- unpack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, [1142](#)
- unpack\_nas\_SlqsGetLTECphyCAInfo\_t, [893](#)
  - LTECphyCAInfo, [894](#)
  - Tlvresult, [894](#)
- unpack\_omaDmConfigTlv\_t, [914](#)
  - alertmsg, [915](#)
  - alertmsglength, [915](#)
  - state, [915](#)
  - userInputReq, [915](#)
  - userInputTimeout, [915](#)
- unpack\_omaDmFotaTlv\_t, [915](#)
  - description, [917](#)
  - descriptionlength, [917](#)
  - fwloadsize, [917](#)
  - fwloadComplete, [917](#)
  - namelength, [917](#)
  - package\_name, [917](#)
  - sessionType, [917](#)
  - severity, [917](#)
  - state, [917](#)
  - updateCompleteStatus, [917](#)
  - userInputReq, [917](#)
  - userInputTimeout, [917](#)
  - version, [917](#)
  - versionlength, [917](#)
- unpack\_omaDmNotificationsTlv\_t, [917](#)
  - notification, [917](#)
  - sessionStatus, [917](#)
- unpack\_qmi\_t, [917](#)
  - msgid, [918](#)
  - type, [918](#)
- xid, [918](#)
- unpack\_qos\_IPv4Addr\_t, [918](#)
  - addr, [919](#)
  - subnetMask, [919](#)
- unpack\_qos\_IPv6Addr\_t, [919](#)
  - addr, [919](#)
  - prefixLen, [919](#)
- unpack\_qos\_IPv6TrafCls\_t, [919](#)
  - mask, [920](#)
  - val, [920](#)
- unpack\_qos\_Port\_t, [920](#)
  - port, [921](#)
  - range, [921](#)
- unpack\_qos\_QosFlowInfo\_t, [921](#)
  - BearerID, [922](#)
  - is\_RxQFlowGranted\_Available, [922](#)
  - is\_TxQFlowGranted\_Available, [922](#)
  - NumRxFilters, [922](#)
  - NumTxFilters, [922](#)
  - QFlowState, [922](#)
  - RxQFilter, [922](#)
  - RxQFlowGranted, [922](#)
  - TxQFilter, [922](#)
  - TxQFlowGranted, [922](#)
- unpack\_qos\_QosFlowInfoState\_t, [922](#)
  - id, [923](#)
  - isNewFlow, [923](#)
  - state, [923](#)
- unpack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, [1557](#)
- unpack\_qos\_SLQSQosGetNetworkStatus\_t, [923](#)
  - NWQoSStatus, [923](#)
- unpack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, [1558](#)
- unpack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [923](#)
  - ambr\_dl, [924](#)
  - ambr\_dl\_ext, [924](#)
  - ambr\_dl\_ext2, [924](#)
  - ambr\_ul, [924](#)
  - ambr\_ul\_ext, [924](#)
  - ambr\_ul\_ext2, [924](#)
  - apnId, [924](#)
- unpack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, [1558](#)
- unpack\_qos\_SLQSQosSwiReadDataStats\_t, [925](#)
  - apnId, [926](#)
  - numQosFlow, [926](#)
  - qosFlow, [926](#)
  - total\_rx\_bytes, [926](#)
  - total\_rx\_pkt, [926](#)
  - total\_tx\_bytes, [926](#)
  - total\_tx\_pkt, [926](#)
- unpack\_qos\_SLQSSetQosEventCallback
  - qos.h, [1559](#)
- unpack\_qos\_SLQSSetQosEventCallback\_ind
  - qos.h, [1559](#)
- unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, [926](#)

- NumFlows, [926](#)
- QosFlowInfo, [926](#)
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind
  - qos.h, [1559](#)
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, [926](#)
  - status, [927](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind
  - qos.h, [1560](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, [927](#)
  - event, [927](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind
  - qos.h, [1561](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [927](#)
  - event, [928](#)
  - id, [928](#)
  - reason, [928](#)
  - status, [928](#)
- unpack\_qos\_Tos\_t, [937](#)
  - mask, [937](#)
  - val, [937](#)
- unpack\_qos\_dataRate\_t, [918](#)
  - dataRateMax, [918](#)
  - guaranteedRate, [918](#)
- unpack\_qos\_pktErrRate\_t, [920](#)
  - exponent, [920](#)
  - multiplier, [920](#)
- unpack\_qos\_swiQosFilter\_t, [929](#)
  - EspSpi, [931](#)
  - IPv4DstAddr, [931](#)
  - IPv4SrcAddr, [931](#)
  - IPv4Tos, [931](#)
  - IPv6DstAddr, [931](#)
  - IPv6Label, [931](#)
  - IPv6SrcAddr, [931](#)
  - IPv6TrafCls, [931](#)
  - Id, [931](#)
  - index, [931](#)
  - is\_EspSpi\_Available, [931](#)
  - is\_IPv4DstAddr\_Available, [931](#)
  - is\_IPv4SrcAddr\_Available, [931](#)
  - is\_IPv4Tos\_Available, [931](#)
  - is\_IPv6DstAddr\_Available, [931](#)
  - is\_IPv6Label\_Available, [931](#)
  - is\_IPv6SrcAddr\_Available, [931](#)
  - is\_IPv6TrafCls\_Available, [931](#)
  - is\_Id\_Available, [931](#)
  - is\_NxtHdrProto\_Available, [931](#)
  - is\_Precedence\_Available, [931](#)
  - is\_TCPDstPort\_Available, [931](#)
  - is\_TCPSrcPort\_Available, [931](#)
  - is-TranDstPort\_Available, [932](#)
  - is-TranSrcPort\_Available, [932](#)
  - is\_UDPDstPort\_Available, [932](#)
  - is\_UDPSrcPort\_Available, [932](#)
  - NxtHdrProto, [932](#)
  - Precedence, [932](#)
  - TCPDstPort, [932](#)
  - TCPSrcPort, [932](#)
  - TranDstPort, [932](#)
  - TranSrcPort, [932](#)
  - UDPDstPort, [932](#)
  - UDPSrcPort, [932](#)
  - version, [932](#)
- unpack\_qos\_swiQosFlow\_t, [932](#)
  - DataRate, [935](#)
  - index, [935](#)
  - is\_DataRate\_Available, [935](#)
  - is\_Jitter\_Available, [935](#)
  - is\_Latency\_Available, [935](#)
  - is\_LteQci\_Available, [935](#)
  - is\_MaxAllowedPktSz\_Available, [935](#)
  - is\_MinPolicedPktSz\_Available, [935](#)
  - is\_PktErrRate\_Available, [935](#)
  - is\_ProfileId3GPP2\_Available, [935](#)
  - is-TokenBucket\_Available, [935](#)
  - is\_TrafficClass\_Available, [935](#)
  - is\_val\_3GPP2Pri\_Available, [935](#)
  - is\_val\_3GPPImCn\_Available, [935](#)
  - is\_val\_3GPPSigInd\_Available, [935](#)
  - Jitter, [935](#)
  - Latency, [935](#)
  - LteQci, [936](#)
  - MaxAllowedPktSz, [936](#)
  - MinPolicedPktSz, [936](#)
  - PktErrRate, [936](#)
  - ProfileId3GPP2, [936](#)
  - TokenBucket, [936](#)
  - TrafficClass, [936](#)
  - val\_3GPP2Pri, [936](#)
  - val\_3GPPImCn, [936](#)
  - val\_3GPPResResidualBER, [936](#)
  - val\_3GPPSigInd, [936](#)
  - val\_3GPPTraHdlPri, [936](#)
- unpack\_qos\_tokenBucket\_t, [936](#)
  - bucketSz, [936](#)
  - peakRate, [936](#)
  - tokenRate, [936](#)
- unpack\_result\_code\_only
  - common.h, [1073](#)
- unpack\_sms\_SLQSDeleteSMS
  - sms.h, [1568](#)
- unpack\_sms\_SLQSDeleteSMS\_t, [940](#)
- unpack\_sms\_SLQSGetSMS
  - sms.h, [1569](#)
- unpack\_sms\_SLQSGetSMS\_t, [940](#)
  - message, [940](#)
  - messageFormat, [940](#)
  - messageSize, [940](#)
  - messageTag, [940](#)
- unpack\_sms\_SLQSGetSMSList
  - sms.h, [1569](#)
- unpack\_sms\_SLQSGetSMSList\_t, [940](#)
  - messageList, [941](#)
  - messageListSize, [941](#)
- unpack\_sms\_SLQSModifySMSStatus

- sms.h, [1569](#)
- unpack\_sms\_SLQSMODIFYSMSStatus\_t, [941](#)
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind
  - sms.h, [1570](#)
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, [941](#)
  - messageMode, [941](#)
  - storageType, [941](#)
- unpack\_sms\_SendSMS
  - sms.h, [1567](#)
- unpack\_sms\_SendSMS\_t, [938](#)
  - messageFailureCode, [938](#)
  - messageID, [938](#)
- unpack\_sms\_SetNewSMSCallback
  - sms.h, [1568](#)
- unpack\_sms\_SetNewSMSCallback\_ind
  - sms.h, [1568](#)
- unpack\_sms\_SetNewSMSCallback\_ind\_t, [938](#)
  - ETWSTlv, [939](#)
  - IMSTlv, [939](#)
  - MMTlv, [939](#)
  - NewMMTlv, [939](#)
  - SMSCTlv, [939](#)
  - TRMessageTlv, [939](#)
- unpack\_sms\_SetNewSMSCallback\_t, [939](#)
- unpack\_swiloc\_SwiLocGetAutoStart
  - swiloc.h, [1572](#)
- unpack\_swiloc\_SwiLocGetAutoStart\_t, [942](#)
  - fix\_rate, [943](#)
  - fix\_rate\_reported, [943](#)
  - fix\_type, [943](#)
  - fix\_type\_reported, [943](#)
  - function, [943](#)
  - function\_reported, [943](#)
  - max\_dist, [943](#)
  - max\_dist\_reported, [943](#)
  - max\_time, [943](#)
  - max\_time\_reported, [943](#)
- unpack\_swiloc\_SwiLocSetAutoStart
  - swiloc.h, [1573](#)
- unpack\_swiuma\_SLQSOMADMAAlertCallback
  - swiuma.h, [1578](#)
- unpack\_swiuma\_SLQSOMADMAAlertCallback\_ind
  - swiuma.h, [1579](#)
- unpack\_swiuma\_SLQSOMADMAAlertCallback\_ind\_t, [943](#)
  - eventType, [944](#)
- unpack\_swiuma\_SLQSOMADMCancelSession
  - swiuma.h, [1579](#)
- unpack\_swiuma\_SLQSOMADMGetSessionInfo
  - swiuma.h, [1580](#)
- unpack\_swiuma\_SLQSOMADMGetSessionInfo\_t, [944](#)
  - Date, [946](#)
  - DateLength, [946](#)
  - PkgDescLength, [946](#)
  - PkgDescription, [946](#)
  - PkgName, [946](#)
  - PkgNameLength, [946](#)
  - RetryCount, [946](#)
  - SessionState, [946](#)
  - SessionType, [946](#)
  - Severity, [946](#)
  - Source, [946](#)
  - SourceLength, [946](#)
  - Status, [946](#)
  - Time, [947](#)
  - TimeLength, [947](#)
  - UpdateCompleteStatus, [947](#)
- unpack\_swiuma\_SLQSOMADMGetSettings
  - swiuma.h, [1580](#)
- unpack\_swiuma\_SLQSOMADMGetSettings\_t, [947](#)
  - Autosdm, [948](#)
  - FOTAdownload, [948](#)
  - FwAutoCheck, [948](#)
- unpack\_swiuma\_SLQSOMADMSelectSelection
  - swiuma.h, [1581](#)
- unpack\_swiuma\_SLQSOMADMSetSettings
  - swiuma.h, [1581](#)
- unpack\_swiuma\_SLQSOMADMStartSession
  - swiuma.h, [1581](#)
- unpack\_swiuma\_SLQSOMADMStartSession\_t, [948](#)
  - FwAvailability, [948](#)
- unpack\_uim\_ChangePin
  - uim.h, [1587](#)
- unpack\_uim\_ChangePin\_t, [949](#)
  - pEncryptedPIN1, [949](#)
  - pIndicationToken, [949](#)
  - pRemainingRetries, [949](#)
  - Tlvresult, [949](#)
- unpack\_uim\_GetCardStatus
  - uim.h, [1588](#)
- unpack\_uim\_GetCardStatus\_t, [949](#)
  - pCardStatus, [950](#)
  - pHotSwapStatus, [950](#)
  - Tlvresult, [950](#)
- unpack\_uim\_ReadTransparent
  - uim.h, [1588](#)
- unpack\_uim\_ReadTransparent\_t, [950](#)
  - pCardResult, [951](#)
  - pEncryptedData, [951](#)
  - pIndicationToken, [951](#)
  - pReadResult, [951](#)
  - Tlvresult, [951](#)
- unpack\_uim\_SLQSUIEventRegister
  - uim.h, [1589](#)
- unpack\_uim\_SLQSUIEventRegister\_t, [952](#)
  - eventMask, [952](#)
- unpack\_uim\_SLQSUIMGetSlotsStatus
  - uim.h, [1589](#)
- unpack\_uim\_SLQSUIMGetSlotsStatus\_t, [952](#)
  - pNumberOfPhySlot, [953](#)
  - pUimSlotsStatus, [953](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind
  - uim.h, [1590](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind\_t, [953](#)

- unpack\_uim\_SLQSUIMSwitchSlot
  - uim.h, [1590](#)
- unpack\_uim\_SetPinProtection
  - uim.h, [1588](#)
- unpack\_uim\_SetPinProtection\_t, [951](#)
  - pEncryptedPIN1, [951](#)
  - pIndicationToken, [951](#)
  - pRemainingRetries, [951](#)
  - Tlvresult, [951](#)
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind
  - uim.h, [1589](#)
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, [952](#)
  - bNumberOfPhySlots, [952](#)
  - slotsstatusChange, [952](#)
- unpack\_uim\_UnblockPin
  - uim.h, [1590](#)
- unpack\_uim\_UnblockPin\_t, [953](#)
  - pEncryptedPIN1, [954](#)
  - pIndicationToken, [954](#)
  - pRemainingRetries, [954](#)
  - Tlvresult, [954](#)
- unpack\_uim\_VerifyPin
  - uim.h, [1591](#)
- unpack\_uim\_VerifyPin\_t, [954](#)
  - pEncryptedPIN1, [955](#)
  - pIndicationToken, [955](#)
  - pRemainingRetries, [955](#)
  - Tlvresult, [955](#)
- unpack\_wds\_GetConnectionRate
  - wds.h, [1608](#)
- unpack\_wds\_GetConnectionRate\_t, [955](#)
  - currentChannelRXRate, [955](#)
  - currentChannelTXRate, [955](#)
  - maxChannelRXRate, [955](#)
  - maxChannelTXRate, [955](#)
- unpack\_wds\_GetDefaultProfile
  - wds.h, [1608](#)
- unpack\_wds\_GetDefaultProfile\_t, [955](#)
  - apnname, [956](#)
  - apnsize, [956](#)
  - auth, [956](#)
  - ipaddr, [956](#)
  - ipaddrv6, [956](#)
  - name, [956](#)
  - namesize, [956](#)
  - pdptype, [956](#)
  - pridns, [956](#)
  - pridnsv6, [956](#)
  - secdns, [956](#)
  - secdnsv6, [956](#)
  - username, [956](#)
  - usersize, [957](#)
- unpack\_wds\_GetDefaultProfileNum
  - wds.h, [1608](#)
- unpack\_wds\_GetDefaultProfileNum\_t, [957](#)
  - index, [957](#)
- unpack\_wds\_GetDormancyState
  - wds.h, [1609](#)
- unpack\_wds\_GetDormancyState\_t, [957](#)
  - dormancyState, [957](#)
- unpack\_wds\_GetLastMobileIPError
  - wds.h, [1609](#)
- unpack\_wds\_GetLastMobileIPError\_t, [957](#)
  - error, [957](#)
- unpack\_wds\_GetMobileIP
  - wds.h, [1609](#)
- unpack\_wds\_GetMobileIP\_t, [958](#)
  - mipMode, [958](#)
- unpack\_wds\_GetMobileIPProfile
  - wds.h, [1610](#)
- unpack\_wds\_GetMobileIPProfile\_t, [958](#)
  - AAASPI, [958](#)
  - AAASState, [958](#)
  - address, [959](#)
  - enabled, [959](#)
  - HASPI, [959](#)
  - HASState, [959](#)
  - NAI, [959](#)
  - naiSize, [959](#)
  - primaryHA, [959](#)
  - revTunneling, [959](#)
  - secondaryHA, [959](#)
- unpack\_wds\_GetPacketStatus
  - wds.h, [1610](#)
- unpack\_wds\_GetPacketStatus\_t, [959](#)
  - rXDroppedCount, [960](#)
  - rXOKBytesLastCall, [960](#)
  - rXOkBytesCount, [960](#)
  - rXPacketErrors, [960](#)
  - rXPacketOverflows, [960](#)
  - rXPacketSuccesses, [960](#)
  - tXDroppedCount, [960](#)
  - tXOKBytesLastCall, [960](#)
  - tXOkBytesCount, [960](#)
  - tXPacketErrors, [960](#)
  - tXPacketOverflows, [960](#)
  - tXPacketSuccesses, [960](#)
- unpack\_wds\_GetSessionDuration
  - wds.h, [1610](#)
- unpack\_wds\_GetSessionDuration\_t, [960](#)
  - callDuration, [960](#)
- unpack\_wds\_GetSessionState
  - wds.h, [1611](#)
- unpack\_wds\_GetSessionState\_t, [960](#)
  - connectionStatus, [961](#)
- unpack\_wds\_RMSetTransferStatistics
  - wds.h, [1611](#)
- unpack\_wds\_RMSetTransferStatistics\_t, [961](#)
  - pCreateProfileOut, [961](#)
  - pProfileID, [961](#)
  - Tlvresult, [961](#)
- unpack\_wds\_SLQSCreateProfile
  - wds.h, [1612](#)
- unpack\_wds\_SLQSCreateProfile\_t, [961](#)
  - pCreateProfileOut, [961](#)
  - pProfileID, [961](#)
  - Tlvresult, [961](#)
- unpack\_wds\_SLQSDeleteProfile

- wds.h, [1613](#)
- unpack\_wds\_SLQSDeleteProfile\_t, [961](#)
  - extendedErrorCode, [962](#)
- unpack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, [1613](#)
- unpack\_wds\_SLQSGet3GPPConfigItem\_t, [962](#)
  - \_3gppRelease, [962](#)
  - defaultPDNEnabled, [962](#)
  - profileList, [963](#)
- unpack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, [1613](#)
- unpack\_wds\_SLQSGetCurrDataSystemStat\_t, [963](#)
  - currNetworkInfo, [963](#)
  - networkInfoLen, [963](#)
  - prefNetwork, [963](#)
- unpack\_wds\_SLQSGetDUNCallInfo
  - wds.h, [1614](#)
- unpack\_wds\_SLQSGetDUNCallInfo\_t, [964](#)
  - callEndReason, [964](#)
  - channelRate, [964](#)
  - connectionStatus, [964](#)
  - dataBearerTech, [964](#)
  - dormancyStatus, [964](#)
  - lastCallDataBearerTech, [964](#)
  - mdmCallDurationActive, [965](#)
  - rxOKBytesCount, [965](#)
  - txOKBytesCount, [965](#)
- unpack\_wds\_SLQSGetDataBearerTechnology
  - wds.h, [1614](#)
- unpack\_wds\_SLQSGetDataBearerTechnology\_t, [963](#)
  - curDataBearerTechnology, [963](#)
  - dataBearerMask, [963](#)
  - lastCallDataBearerTechnology, [964](#)
- unpack\_wds\_SLQSGetProfileSettings
  - wds.h, [1615](#)
- unpack\_wds\_SLQSGetProfileSettings\_t, [965](#)
  - pProfileSettings, [965](#)
  - ProfileType, [965](#)
  - Tlvresult, [965](#)
- unpack\_wds\_SLQSGetRuntimeSettings
  - wds.h, [1615](#)
- unpack\_wds\_SLQSGetRuntimeSettings\_t, [965](#)
  - APNName, [966](#)
  - Authentication, [966](#)
  - DomainList, [966](#)
  - GPRSGrantedQoS, [966](#)
  - GWAddressV4, [966](#)
  - IMCNflag, [966](#)
  - IPFamilyPreference, [966](#)
  - IPv6AddrInfo, [967](#)
  - IPv6GWAddrInfo, [967](#)
  - IPv4, [966](#)
  - Mtu, [967](#)
  - PDPTType, [967](#)
  - PrimaryDNSV4, [967](#)
  - PrimaryDNSV6, [967](#)
  - ProfileID, [967](#)
  - ProfileName, [967](#)
  - SecondaryDNSV4, [967](#)
  - SecondaryDNSV6, [967](#)
  - ServerAddrList, [967](#)
  - SubnetMaskV4, [967](#)
  - Technology, [967](#)
  - UMTSGrantedQoS, [967](#)
  - Username, [967](#)
- unpack\_wds\_SLQSModifyProfile
  - wds.h, [1615](#)
- unpack\_wds\_SLQSModifyProfile\_t, [967](#)
  - pExtErrorCode, [967](#)
- unpack\_wds\_SLQSSetDHCPv4ClientConfig
  - wds.h, [1617](#)
- unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [970](#)
  - pHwConfig, [970](#)
- unpack\_wds\_SLQSSet3GPPConfigItem
  - wds.h, [1615](#)
- unpack\_wds\_SLQSSetIPFamilyPreference
  - wds.h, [1616](#)
- unpack\_wds\_SLQSSetIPFamilyPreference\_t, [967](#)
  - Tlvresult, [968](#)
- unpack\_wds\_SLQSSetPacketSrvStatusCallback
  - wds.h, [1616](#)
- unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [968](#)
  - bearerID, [968](#)
  - conn\_status, [968](#)
  - ipFamily, [968](#)
  - reconfigReqd, [969](#)
  - sessionEndReason, [969](#)
  - techName, [969](#)
  - verboseSessnEndReason, [969](#)
  - verboseSessnEndReasonType, [969](#)
- unpack\_wds\_SLQSSetWdsEventCallback
  - wds.h, [1616](#)
- unpack\_wds\_SLQSSetWdsEventCallback\_ind
  - wds.h, [1617](#)
- unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [969](#)
  - currDBTechAvail, [970](#)
  - currNWInfo, [970](#)
  - dBTechAvail, [970](#)
  - dBTechnology, [970](#)
  - dataSysStatAvail, [970](#)
  - dormancyStatAvail, [970](#)
  - dormancyStatus, [970](#)
  - mipStatus, [970](#)
  - mipstatAvail, [970](#)
  - netInfoLen, [970](#)
  - prefNetwork, [970](#)
  - ratMask, [970](#)
  - rx\_bytes, [970](#)
  - rx\_pkts, [970](#)
  - soMask, [970](#)
  - tx\_bytes, [970](#)
  - tx\_pkts, [970](#)
  - xferStatAvail, [970](#)
- unpack\_wds\_SLQSStartDataSession
  - wds.h, [1617](#)
- unpack\_wds\_SLQSStartDataSession\_t, [971](#)

- pFailureReason, [971](#)
  - pVerboseFailReasonType, [971](#)
  - pVerboseFailureReason, [971](#)
  - psid, [971](#)
- unpack\_wds\_SLQSSStopDataSession
  - wds.h, [1618](#)
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings
  - wds.h, [1618](#)
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [971](#)
  - apnName, [972](#)
  - bearerId, [972](#)
  - contextId, [972](#)
  - ipv4Address, [972](#)
  - ipv4GWAddress, [972](#)
  - ipv6Address, [972](#)
  - ipv6GWAddress, [972](#)
- unpack\_wds\_SetDefaultProfile
  - wds.h, [1611](#)
- unpack\_wds\_SetDefaultProfileNum
  - wds.h, [1612](#)
- unpack\_wds\_SetMobileIPProfile
  - wds.h, [1612](#)
- unpack\_wds\_SetMobileIPProfile\_t, [961](#)
- UnpackQmiProfileInfo
  - wds.h, [1595](#)
- unpackWdsProfileParam, [973](#)
  - SlqsProfile3GPP, [973](#)
  - SlqsProfile3GPP2, [973](#)
- upLink
  - NSSAudioCtrl, [511](#)
- UpdateCompleteStatus
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [947](#)
- updateCompleteStatus
  - omaDmFotaTlv, [516](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- upgrade\_mc77xx\_fw
  - qaGobiApiFms.h, [1299](#)
- UpgradeFirmware2k
  - qaGobiApiFms.h, [1300](#)
- upinRetries
  - slotInf, [712](#)
  - slotInfo, [714](#)
  - uim\_slotInfo, [807](#)
- upinState
  - slotInf, [713](#)
  - slotInfo, [714](#)
  - uim\_slotInfo, [808](#)
- UpkQmiCbkCatEventReportInd
  - qaCbkCatEventReportInd.h, [1150](#)
- UpkQmiCbkSwiOmaDmEventReportInd
  - qaCbkSwiOmaDmEventReportInd.h, [1151](#)
- UpkQmiCbkSwiOmaDmEventReportIndExt
  - qaCbkSwiOmaDmEventReportInd.h, [1151](#)
- UpkQmiNasGetRFBandInfo
  - qaNasGetRFBandInfo.h, [1545](#)
- UpkQmiNasPerformNetworkScan
  - qaNasPerformNetworkScan.h, [1546](#)
- upukRetries
  - slotInf, [713](#)
  - slotInfo, [714](#)
  - uim\_slotInfo, [808](#)
- usageMask
  - loc\_sensorDataUsage, [344](#)
  - sensorDataUsage\_s, [676](#)
  - t\_sensor, [775](#)
- User Identity Module Service (UIM), [46](#)
- userData
  - SMSAsyncRawSend\_s, [732](#)
- userInputReq
  - omaDmConfigTlv, [512](#)
  - omaDmConfigTlvExt, [514](#)
  - omaDmFotaTlv, [516](#)
  - unpack\_omaDmConfigTlv\_t, [915](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- userInputTimeout
  - omaDmConfigTlv, [512](#)
  - omaDmConfigTlvExt, [514](#)
  - omaDmFotaTlv, [516](#)
  - omaDmFotaTlvExt, [518](#)
  - unpack\_omaDmConfigTlv\_t, [915](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- Username
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [967](#)
- username
  - unpack\_wds\_GetDefaultProfile\_t, [956](#)
- usersize
  - unpack\_wds\_GetDefaultProfile\_t, [957](#)
- ussDCS
  - USSInfo, [978](#)
- ussData
  - USSInfo, [978](#)
- ussLen
  - USSInfo, [978](#)
- uusInfo
  - allCallsUUSInfo, [98](#)
- v4sessionId
  - qaQmiInterfaceInfo, [615](#)
  - ssdatasession\_params, [752](#)
  - WdsRunTimeSettings, [1065](#)
- v6sessionId
  - qaQmiInterfaceInfo, [615](#)
  - ssdatasession\_params, [752](#)
  - WdsRunTimeSettings, [1065](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATA
  - qaGobiApiVoice.h, [1485](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATA\_CIRCUITASYNC
  - qaGobiApiVoice.h, [1486](#)
- VOICE\_SUPS\_SRV\_CLASS\_DATA\_CIRCUITSYNC
  - qaGobiApiVoice.h, [1486](#)
- VOICE\_SUPS\_SRV\_CLASS\_FAX
  - qaGobiApiVoice.h, [1485](#)
- VOICE\_SUPS\_SRV\_CLASS\_NONE
  - qaGobiApiVoice.h, [1485](#)
- VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS

- qaGobiApiVoice.h, [1486](#)
- VOICE\_SUPS\_SRV\_CLASS\_PADACCESS
  - qaGobiApiVoice.h, [1486](#)
- VOICE\_SUPS\_SRV\_CLASS\_SMS
  - qaGobiApiVoice.h, [1486](#)
- VOICE\_SUPS\_SRV\_CLASS\_VOICE
  - qaGobiApiVoice.h, [1485](#)
- VDOP
  - loc\_precisionDilution, [343](#)
  - precisionDilution\_s, [598](#)
- VOICE\_SRV
  - qaGobiApiCbK.h, [1165](#)
- val
  - IPv6TrafCls, [306](#)
  - Tos, [789](#)
  - unpack\_qos\_IPv6TrafCls\_t, [920](#)
  - unpack\_qos\_Tos\_t, [937](#)
- val\_3GPP2Pri
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- val\_3GPPImCn
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- val\_3GPPResResidualBER
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- val\_3GPPSigInd
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- val\_3GPPTraHdlPri
  - unpack\_qos\_swiQosFlow\_t, [936](#)
- ValidMask
  - GPSSStateInfo, [266](#)
- validMask
  - satelliteInfo, [674](#)
- ValidateSPC
  - qaGobiApiDms.h, [1283](#)
- ValidityCW0
  - LteCQIParm, [359](#)
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, [911](#)
- ValidityCW1
  - LteCQIParm, [359](#)
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, [911](#)
- Value
  - GetM2MSpkrGainResp, [254](#)
  - SetM2MSpkrGainReq, [698](#)
- value\_length
  - custSettingInfo, [184](#)
  - DMScustSettingInfo, [202](#)
  - pack\_dms\_SetCustFeaturesV2\_t, [523](#)
  - setCustomSettingV2, [688](#)
- verbFailReason
  - ssdatasession\_params, [752](#)
- verbFailReasonType
  - ssdatasession\_params, [752](#)
- verboseSessnEndReason
  - \_packetSrvStatus, [62](#)
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [969](#)
- verboseSessnEndReasonType
  - \_packetSrvStatus, [62](#)
- unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [969](#)
- verifyLeft
  - personalizationStatus, [588](#)
  - remainingRetries, [657](#)
  - uim\_remainingRetries, [804](#)
- verifyPIN
  - pack\_uim\_VerifyPin\_t, [566](#)
  - UIMVerifyPinReq, [832](#)
- verifyUIMPIN, [980](#)
  - pinID, [980](#)
  - pinLen, [980](#)
  - pinVal, [980](#)
- version
  - omaDmFotaTlv, [516](#)
  - omaDmFotaTlvExt, [518](#)
  - swiQosFilter, [767](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
  - unpack\_qos\_swiQosFilter\_t, [932](#)
- versionlength
  - omaDmFotaTlv, [516](#)
  - omaDmFotaTlvExt, [518](#)
  - unpack\_omaDmFotaTlv\_t, [917](#)
- VerticalUncertainty
  - GPSSStateInfo, [266](#)
- VirtStream
  - protocolSubtypeElement, [614](#)
- Voice Service (VOICE), [44](#)
- voiceALSSelectLineInfo, [980](#)
  - lineValue, [981](#)
- voiceALSSetLineSwitchInfo, [981](#)
  - switchOption, [981](#)
- voiceAnswerCall, [981](#)
  - pCallId, [982](#)
- voiceBindSubscriptionInfo, [982](#)
  - subsType, [982](#)
- voiceBurstDTMFInfo, [982](#)
  - BurstDTMFInfo, [982](#)
  - pBurstDTMFLengths, [982](#)
- voiceCallInfoReq, [983](#)
  - callID, [983](#)
- voiceCallInfoResp, [983](#)
  - pAlertType, [985](#)
  - pAlertingPattern, [985](#)
  - pAlphaIDInfo, [985](#)
  - pCallInfo, [985](#)
  - pConnectNumInfo, [985](#)
  - pDiagInfo, [985](#)
  - pOTASPStatus, [985](#)
  - pRemotePartyName, [985](#)
  - pRemotePartyNum, [985](#)
  - pSrvOpt, [985](#)
  - pUUSInfo, [985](#)
  - pVoicePrivacy, [986](#)
- voiceCallRequestParams, [986](#)
  - callNumber, [987](#)
  - pCLIRType, [987](#)
  - pCUGInfo, [987](#)

- pCallPartySubAdd, [987](#)
  - pCallType, [987](#)
  - pEmergencyCategory, [987](#)
  - pSvcType, [987](#)
  - pUUSInfo, [987](#)
- voiceCallResponseParams, [987](#)
  - pAlphaInfo, [988](#)
  - pCCResultType, [988](#)
  - pCCSUPSType, [988](#)
  - pCallID, [988](#)
- voiceContDTMFInfo, [988](#)
  - DTMFdigit, [989](#)
  - pCallID, [989](#)
- voiceDTMFEventInfo, [989](#)
  - DTMFInformation, [990](#)
  - pOffLength, [990](#)
  - pOnLength, [990](#)
- voiceFlashInfo, [990](#)
  - pCallID, [990](#)
  - pFlashPayLd, [990](#)
  - pFlashType, [990](#)
- voiceGetAllCallInfo, [990](#)
  - pArrAlertingPattern, [993](#)
  - pArrAlertingType, [993](#)
  - pArrAlphaID, [993](#)
  - pArrCallEndReason, [993](#)
  - pArrCallInfo, [993](#)
  - pArrCalledPartyNum, [993](#)
  - pArrConnectPartyNum, [993](#)
  - pArrDiagInfo, [993](#)
  - pArrRedirPartyNum, [993](#)
  - pArrRemotePartyName, [993](#)
  - pArrRemotePartyNum, [993](#)
  - pArrSvcOption, [993](#)
  - pArrUUSInfo, [993](#)
  - pOTASPStatus, [993](#)
  - pVoicePrivacy, [993](#)
- voiceGetCLIPResp, [999](#)
  - pAlphaInfo, [1000](#)
  - pCCResType, [1000](#)
  - pCCSUPSType, [1000](#)
  - pCLIPResp, [1000](#)
  - pCallID, [1000](#)
  - pFailCause, [1000](#)
- voiceGetCLIRResp, [1000](#)
  - pAlphaInfo, [1001](#)
  - pCCResType, [1001](#)
  - pCCSUPSType, [1001](#)
  - pCLIRResp, [1001](#)
  - pCallID, [1001](#)
  - pFailCause, [1001](#)
- voiceGetCNAPResp, [1001](#)
  - pAlphaInfo, [1002](#)
  - pCCResType, [1002](#)
  - pCCSUPSType, [1002](#)
  - pCNAPResp, [1002](#)
  - pCallID, [1002](#)
  - pFailCause, [1002](#)
- voiceGetCOLPResp, [1002](#)
  - pAlphaInfo, [1003](#)
  - pCCResType, [1004](#)
  - pCCSUPSType, [1004](#)
  - pCOLPResp, [1004](#)
  - pCallID, [1003](#)
  - pFailCause, [1004](#)
- voiceGetCOLRResp, [1004](#)
  - pAlphaInfo, [1005](#)
  - pCCResType, [1005](#)
  - pCCSUPSType, [1005](#)
  - pCOLRResp, [1005](#)
  - pCallID, [1005](#)
  - pFailCause, [1005](#)
- voiceGetCallBarringReq, [993](#)
  - pSvcClass, [994](#)
  - reason, [994](#)
- voiceGetCallBarringResp, [994](#)
  - pAlphaInfo, [995](#)
  - pCCResType, [995](#)
  - pCCSUPSType, [995](#)
  - pCallID, [995](#)
  - pFailCause, [995](#)
  - pSvcClass, [995](#)
- voiceGetCallFWReq, [995](#)
  - pSvcClass, [996](#)
  - Reason, [996](#)
- voiceGetCallFWResp, [996](#)
  - pAlphaInfo, [997](#)
  - pCCResType, [997](#)
  - pCCSUPSType, [997](#)
  - pCallID, [997](#)
  - pFailCause, [997](#)
  - pGetCallFWExtInfo, [997](#)
  - pGetCallFWInfo, [997](#)
- voiceGetCallWaitInfo, [997](#)
  - pAlphaInfo, [998](#)
  - pCCResType, [999](#)
  - pCCSUPSType, [999](#)
  - pCallID, [999](#)
  - pFailCause, [999](#)
  - pSvcClass, [999](#)
- voiceGetConfigReq, [1005](#)
  - pAMRStatus, [1006](#)
  - pAirTimer, [1006](#)
  - pAutoAnswer, [1006](#)
  - pNamID, [1006](#)
  - pPrefVoicePrivacy, [1006](#)
  - pPrefVoiceSO, [1006](#)
  - pRoamTimer, [1006](#)
  - pTTYMode, [1007](#)
  - pVoiceDomainPref, [1007](#)
- voiceGetConfigResp, [1007](#)
  - pAirTimerCnt, [1008](#)
  - pAutoAnswerStat, [1008](#)
  - pCurAMRConfig, [1008](#)
  - pCurPrefVoiceSO, [1008](#)
  - pCurVoiceDomainPref, [1008](#)

- pCurVoicePrivacyPref, [1008](#)
  - pCurrTTYMode, [1008](#)
  - pRoamTimerCnt, [1008](#)
- voiceIndicationRegisterInfo, [1008](#)
  - pRegDTMFEvents, [1009](#)
  - pRegVoicePrivacyEvents, [1009](#)
  - pSuppsNotifEvents, [1009](#)
- voiceInfoRec, [1009](#)
  - callID, [1011](#)
  - pCLIRCause, [1011](#)
  - pCallWaitInd, [1011](#)
  - pCalledPartyInfo, [1011](#)
  - pCallerIDInfo, [1011](#)
  - pCallerNameInfo, [1011](#)
  - pCallingPartyInfo, [1011](#)
  - pConnectNumInfo, [1011](#)
  - pDisplInfo, [1011](#)
  - pExtDisplInfo, [1011](#)
  - pExtDispRecInfo, [1011](#)
  - pLineCtrlInfo, [1011](#)
  - pNSSAudioCtrl, [1011](#)
  - pNSSRelease, [1011](#)
  - pRedirNumInfo, [1011](#)
  - pSignalInfo, [1012](#)
- voiceManageCallsReq, [1012](#)
  - pCallID, [1012](#)
  - SUPSType, [1012](#)
- voiceManageCallsResp, [1012](#)
  - pFailCause, [1013](#)
- VoiceNumber
  - unpack\_dms\_GetVoiceNumber\_t, [854](#)
- voiceNumberSize
  - unpack\_dms\_GetVoiceNumber\_t, [854](#)
- voiceOTASPStatusInfo, [1013](#)
  - callID, [1014](#)
  - OTASPStatus, [1014](#)
- voiceOrigUSSDNoWaitInfo, [1013](#)
  - USSInformation, [1013](#)
- voicePrivacy
  - voicePrivacyInfo, [1015](#)
- voicePrivacyInfo, [1014](#)
  - callID, [1015](#)
  - voicePrivacy, [1015](#)
- voiceSUPSInfo, [1027](#)
  - pAlphaIDInfo, [1029](#)
  - pCLIPstatus, [1029](#)
  - pCLIRstatus, [1029](#)
  - pCNAPstatus, [1029](#)
  - pCOLPstatus, [1029](#)
  - pCOLRstatus, [1029](#)
  - pCallBarPasswd, [1029](#)
  - pCallFWNum, [1029](#)
  - pCallFWTimerVal, [1029](#)
  - pCallFwdInfo, [1029](#)
  - pCallID, [1029](#)
  - pDataSrc, [1029](#)
  - pFailCause, [1029](#)
  - pNewPwdData, [1029](#)
  - pReason, [1029](#)
  - pSvcClass, [1029](#)
  - pUSSInfo, [1029](#)
  - SUPSInformation, [1029](#)
- voiceSUPSNotification, [1029](#)
  - callID, [1031](#)
  - notifType, [1031](#)
  - pCUGIndex, [1031](#)
  - pECTNum, [1031](#)
- voiceSetAllCallStatusCbklInfo, [1015](#)
  - arrCallInfomation, [1017](#)
  - pArrAlertingPattern, [1017](#)
  - pArrAlertingType, [1017](#)
  - pArrAlphaID, [1017](#)
  - pArrCallEndReason, [1017](#)
  - pArrCalledPartyNum, [1017](#)
  - pArrConnectPartyNum, [1017](#)
  - pArrDiagInfo, [1017](#)
  - pArrRedirPartyNum, [1017](#)
  - pArrRemotePartyName, [1017](#)
  - pArrRemotePartyNum, [1017](#)
  - pArrSvcOption, [1017](#)
- voiceSetCallBarringPwdInfo, [1017](#)
  - newPasswd, [1018](#)
  - newPasswdAgain, [1018](#)
  - oldPasswd, [1018](#)
  - Reason, [1018](#)
- voiceSetCallBarringPwdResp, [1018](#)
  - pAlphaIDInfo, [1019](#)
  - pCCResType, [1019](#)
  - pCCSUPSType, [1019](#)
  - pCallID, [1019](#)
  - pFailCause, [1019](#)
- voiceSetConfigReq, [1019](#)
  - pAirTimerConfig, [1020](#)
  - pAutoAnswer, [1021](#)
  - pPrefVoiceDomain, [1021](#)
  - pPrefVoiceSO, [1021](#)
  - pRoamTimerConfig, [1021](#)
  - pTTYMode, [1021](#)
- voiceSetConfigResp, [1021](#)
  - pAirTimerStatus, [1022](#)
  - pAutoAnsStatus, [1022](#)
  - pPrefVoiceSOStatus, [1022](#)
  - pRoamTimerStatus, [1022](#)
  - pTTYConfigStatus, [1022](#)
  - pVoiceDomainPrefStatus, [1022](#)
- voiceSetPrefPrivacy, [1022](#)
  - privacyPref, [1023](#)
- voiceSetSUPSServiceReq, [1023](#)
  - pCallBarringPasswd, [1025](#)
  - pCallForwardingNumber, [1025](#)
  - pCallFwdTypeAndPlan, [1025](#)
  - pServiceClass, [1025](#)
  - pTimerVal, [1025](#)
  - reason, [1025](#)
  - voiceSvc, [1025](#)
- voiceSetSUPSServiceResp, [1025](#)

- pAlphaIDInfo, [1026](#)
  - pCCResultType, [1026](#)
  - pCCSUPSType, [1026](#)
  - pCallID, [1026](#)
  - pFailCause, [1026](#)
- voiceStopContDTMFinfo, [1026](#)
  - callID, [1027](#)
- voiceSvc
  - voiceSetSUPSServiceReq, [1025](#)
- VolValue
  - SetAudioVolTLBConfigReq, [687](#)
- Volume
  - GetAudioProfileResp, [234](#)
  - GetAudioVolTLBConfigReq, [235](#)
  - GetM2MAudioProfileResp, [251](#)
  - SetAudioProfileReq, [686](#)
  - SetAudioVolTLBConfigReq, [687](#)
- voteForInit
  - registerRefresh, [657](#)
- WCDMACellInfo
  - lteWcdmaCellInfo, [378](#)
  - nas\_lteWcdmaCellInfo, [431](#)
- WCDMAECIOThresh, [1032](#)
  - pWCDMAECIOThreshList, [1032](#)
  - WCDMAECIOThreshListLen, [1032](#)
- WCDMAECIOThreshListLen
  - nas\_WCDMAECIOThresh, [462](#)
  - WCDMAECIOThresh, [1032](#)
- WCDMAInfoLTENeighborCell, [1032](#)
  - UMTSLTENbrCell, [1033](#)
  - umtsLTENbrCellLen, [1033](#)
  - wcdmaRRCTest, [1033](#)
- WCDMARSSIThresh, [1037](#)
  - pWCDMARSSIThreshList, [1038](#)
  - WCDMARSSIThreshListLen, [1038](#)
- WCDMARSSIThreshListLen
  - nas\_WCDMARSSIThresh, [464](#)
  - WCDMARSSIThresh, [1038](#)
- WCDMASSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, [907](#)
- WCDMASysInfo, [1038](#)
  - cellId, [1040](#)
  - cellIdValid, [1041](#)
  - hsCallStatus, [1041](#)
  - hsCallStatusValid, [1041](#)
  - hsInd, [1041](#)
  - hsIndValid, [1041](#)
  - lac, [1041](#)
  - lacValid, [1041](#)
  - MCC, [1041](#)
  - MNC, [1041](#)
  - networkIdValid, [1041](#)
  - psc, [1041](#)
  - pscValid, [1041](#)
  - regRejectInfoValid, [1041](#)
  - rejCause, [1041](#)
  - rejectSrvDomain, [1041](#)
  - sysInfoWCDMA, [1041](#)
- WDS\_IsGobiDevice
  - qaGobiApiWds.h, [1544](#)
- WDS\_SRV
  - qaGobiApiCbk.h, [1166](#)
- WDSGetLoopbackData, [1060](#)
  - ByteLoopbackMode, [1061](#)
  - ByteLoopbackMultiplier, [1061](#)
- WDSSWICurrentChannelRates, [1067](#)
  - current\_channel\_rx\_rate, [1068](#)
  - current\_channel\_tx\_rate, [1068](#)
  - max\_channel\_rx\_rate, [1068](#)
  - max\_channel\_tx\_rate, [1068](#)
- WDSSetLoopbackData, [1067](#)
  - pLoopbackMode, [1067](#)
  - pLoopbackMultiplier, [1067](#)
- WORD
  - SwiDataTypes.h, [1571](#)
- wcdmaAmrStat
  - curAMRConfig, [172](#)
- wcdmaCellInfo, [1031](#)
  - cpich\_ecno, [1031](#)
  - cpich\_rscp, [1031](#)
  - psc, [1031](#)
  - srxlev, [1032](#)
- wcdmaLongMsgDecodingParams, [1033](#)
  - Date, [1034](#)
  - plsUDHPresent, [1034](#)
  - pMessage, [1034](#)
  - pPartNum, [1035](#)
  - pReferenceNum, [1035](#)
  - pScAddr, [1035](#)
  - pScAddrLength, [1035](#)
  - pSenderAddr, [1035](#)
  - pSenderAddrLength, [1035](#)
  - pTextMsg, [1035](#)
  - pTextMsgLength, [1035](#)
  - pTotalNum, [1035](#)
  - Time, [1035](#)
- wcdmaMsgDecodingParams, [1035](#)
  - Date, [1036](#)
  - pMessage, [1036](#)
  - pScAddr, [1036](#)
  - pScAddrLength, [1036](#)
  - pSenderAddr, [1036](#)
  - pSenderAddrLength, [1036](#)
  - pTextMsg, [1036](#)
  - pTextMsgLength, [1036](#)
  - Time, [1036](#)
- wcdmaMsgEncodingParams, [1036](#)
  - alphabet, [1037](#)
  - messageSize, [1037](#)
  - pDestAddr, [1037](#)
  - PDUMessage, [1037](#)
  - pTextMsg, [1037](#)
- wcdmaRRCTest
  - nas\_WCDMAInfoLTENeighborCell, [463](#)
  - WCDMAInfoLTENeighborCell, [1033](#)
- wcdmaUARFCN, [1041](#)

- status, [1042](#)
- uarfcn, [1042](#)
- wds.h, [1591](#)
  - PACK\_WDS\_IPV4, [1595](#)
  - PACK\_WDS\_IPV6, [1595](#)
  - pack\_wds\_GetConnectionRate, [1595](#)
  - pack\_wds\_GetDefaultProfile, [1596](#)
  - pack\_wds\_GetDefaultProfileNum, [1596](#)
  - pack\_wds\_GetDormancyState, [1597](#)
  - pack\_wds\_GetLastMobileIPError, [1597](#)
  - pack\_wds\_GetMobileIP, [1597](#)
  - pack\_wds\_GetMobileIPProfile, [1598](#)
  - pack\_wds\_GetPacketStatus, [1598](#)
  - pack\_wds\_GetSessionDuration, [1599](#)
  - pack\_wds\_GetSessionState, [1599](#)
  - pack\_wds\_RMSetTransferStatistics, [1600](#)
  - pack\_wds\_SLQSCreateProfile, [1601](#)
  - pack\_wds\_SLQSDeleteProfile, [1601](#)
  - pack\_wds\_SLQSGet3GPPConfigItem, [1602](#)
  - pack\_wds\_SLQSGetCurrDataSystemStat, [1602](#)
  - pack\_wds\_SLQSGetDUNCallInfo, [1603](#)
  - pack\_wds\_SLQSGetDataBearerTechnology, [1603](#)
  - pack\_wds\_SLQSGetProfileSettings, [1604](#)
  - pack\_wds\_SLQSGetRuntimeSettings, [1604](#)
  - pack\_wds\_SLQSModifyProfile, [1604](#)
  - pack\_wds\_SLQSSetDHCPv4ClientConfig, [1606](#)
  - pack\_wds\_SLQSSet3GPPConfigItem, [1605](#)
  - pack\_wds\_SLQSSetIPFamilyPreference, [1605](#)
  - pack\_wds\_SLQSSetWdsEventCallback, [1606](#)
  - pack\_wds\_SLQSStartDataSession, [1606](#)
  - pack\_wds\_SLQSStopDataSession, [1607](#)
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings, [1607](#)
  - pack\_wds\_SetDefaultProfile, [1600](#)
  - pack\_wds\_SetDefaultProfileNum, [1600](#)
  - pack\_wds\_SetMobileIPProfile, [1601](#)
  - unpack\_wds\_GetConnectionRate, [1608](#)
  - unpack\_wds\_GetDefaultProfile, [1608](#)
  - unpack\_wds\_GetDefaultProfileNum, [1608](#)
  - unpack\_wds\_GetDormancyState, [1609](#)
  - unpack\_wds\_GetLastMobileIPError, [1609](#)
  - unpack\_wds\_GetMobileIP, [1609](#)
  - unpack\_wds\_GetMobileIPProfile, [1610](#)
  - unpack\_wds\_GetPacketStatus, [1610](#)
  - unpack\_wds\_GetSessionDuration, [1610](#)
  - unpack\_wds\_GetSessionState, [1611](#)
  - unpack\_wds\_RMSetTransferStatistics, [1611](#)
  - unpack\_wds\_SLQSCreateProfile, [1612](#)
  - unpack\_wds\_SLQSDeleteProfile, [1613](#)
  - unpack\_wds\_SLQSGet3GPPConfigItem, [1613](#)
  - unpack\_wds\_SLQSGetCurrDataSystemStat, [1613](#)
  - unpack\_wds\_SLQSGetDUNCallInfo, [1614](#)
  - unpack\_wds\_SLQSGetDataBearerTechnology, [1614](#)
  - unpack\_wds\_SLQSGetProfileSettings, [1615](#)
  - unpack\_wds\_SLQSGetRuntimeSettings, [1615](#)
  - unpack\_wds\_SLQSModifyProfile, [1615](#)
  - unpack\_wds\_SLQSSetDHCPv4ClientConfig, [1617](#)
  - unpack\_wds\_SLQSSet3GPPConfigItem, [1615](#)
  - unpack\_wds\_SLQSSetIPFamilyPreference, [1616](#)
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback, [1616](#)
  - unpack\_wds\_SLQSSetWdsEventCallback, [1616](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind, [1617](#)
  - unpack\_wds\_SLQSStartDataSession, [1617](#)
  - unpack\_wds\_SLQSStopDataSession, [1618](#)
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings, [1618](#)
  - unpack\_wds\_SetDefaultProfile, [1611](#)
  - unpack\_wds\_SetDefaultProfileNum, [1612](#)
  - unpack\_wds\_SetMobileIPProfile, [1612](#)
  - UnpackQmiProfileInfo, [1595](#)
- wds\_Domain, [1043](#)
  - domainLen, [1044](#)
  - domainName, [1044](#)
- wds\_DomainNameList, [1044](#)
  - domain, [1044](#)
  - numInstances, [1044](#)
- wds\_GPRSQoS, [1044](#)
  - delayClass, [1045](#)
  - meanThroughputClass, [1045](#)
  - peakThroughputClass, [1045](#)
  - precedenceClass, [1045](#)
  - reliabilityClass, [1045](#)
- wds\_IPV6AddressInfo, [1045](#)
  - IPAddressV6, [1045](#)
  - IPV6PrefixLen, [1045](#)
- wds\_IPV6GWAddressInfo, [1045](#)
  - gwAddressV6, [1046](#)
  - gwV6PrefixLen, [1046](#)
- wds\_PCSCFFQDNAddress, [1046](#)
  - fqdnAddr, [1046](#)
  - fqdnLen, [1046](#)
- wds\_PCSCFFQDNAddressList, [1046](#)
  - numInstances, [1047](#)
  - pcsfFQDNAddress, [1047](#)
- wds\_PCSCFIPv4ServerAddressList, [1047](#)
  - numInstances, [1047](#)
  - pcsfIPv4Addr, [1047](#)
- wds\_ProfileIdentifier, [1047](#)
  - profileIndex, [1048](#)
  - profileType, [1048](#)
- wds\_UMTSMinQoS, [1048](#)
  - deliveryErrSDU, [1050](#)
  - grntDownlinkBitrate, [1050](#)
  - grntUplinkBitrate, [1050](#)
  - maxDownlinkBitrate, [1050](#)
  - maxSDUSize, [1050](#)
  - maxUplinkBitrate, [1050](#)
  - qosDeliveryOrder, [1050](#)
  - resBerRatio, [1050](#)
  - sduErrorRatio, [1051](#)
  - trafficClass, [1051](#)

- trafficPriority, 1051
- transferDelay, 1051
- wds\_currNetworkInfo, 1042
  - NetworkType, 1043
  - RATMask, 1043
  - SOMask, 1043
- wds\_profileInfo, 1048
  - SlqsProfile3GPP, 1048
  - SlqsProfile3GPP2, 1048
- WdsByteTotals, 1051
  - ByteTotalsElmntsV4, 1051
  - ByteTotalsElmntsV6, 1051
  - pV4sessionId, 1051
  - pV6sessionId, 1051
- WdsByteTotalsElmnts, 1052
  - pRXTotalBytes, 1052
  - pTXTotalBytes, 1052
- WdsClientLeaseChange, 1052
  - pEnableNotification, 1052
- WdsConnectionRate, 1052
  - ConnRateElmntsV4, 1053
  - ConnRateElmntsV6, 1053
  - pV4sessionId, 1053
  - pV6sessionId, 1053
- WdsConnectionRateElmnts, 1053
  - pCurrentChannelRXRate, 1054
  - pCurrentChannelTXRate, 1054
  - pMaxChannelRXRate, 1054
  - pMaxChannelTXRate, 1054
- WdsDHCPv4ClientLeaseInd, 1054
  - pIPv4Addr, 1055
  - pLeaseState, 1055
  - pOptList, 1055
  - pProfileId, 1055
- WdsDHCPv4Config, 1055
  - pHwConfig, 1056
  - pProfileId, 1056
  - pRequestOptionList, 1056
- WdsDHCPv4HWConfig, 1056
  - chaddr, 1057
  - chaddrLen, 1057
  - hwType, 1057
- WdsDHCPv4Option, 1057
  - optCode, 1057
  - optVal, 1057
  - optValLen, 1058
- WdsDHCPv4OptionList, 1059
  - numOpt, 1059
  - pOptList, 1059
- WdsDHCPv4ProfileId, 1059
  - profileId, 1059
  - profileType, 1060
- wdsDhcpv4HwConfig, 1056
  - chaddr, 1056
  - chaddrLen, 1056
  - hwType, 1056
- wdsDhcpv4Option, 1058
  - optCode, 1058
  - optVal, 1058
  - optValLen, 1058
- wdsDhcpv4OptionList, 1058
  - numOpt, 1058
  - pOptList, 1058
- wdsDhcpv4ProfileId, 1060
  - profileId, 1060
  - profileType, 1060
- WdsIpAddressInfoReq, 1061
  - ip, 1061
  - pV4sessionId, 1061
  - pV6sessionId, 1061
- WdsPktStatisticsElmnts, 1061
  - pRXDroppedCount, 1062
  - pRXOKBytesLastCall, 1062
  - pRXOkBytesCount, 1062
  - pRXPacketErrors, 1062
  - pRXPacketOverflows, 1063
  - pRXPacketSuccesses, 1063
  - pTXDroppedCount, 1063
  - pTXOKBytesLastCall, 1063
  - pTXOkBytesCount, 1063
  - pTXPacketErrors, 1063
  - pTXPacketOverflows, 1063
  - pTXPacketSuccesses, 1063
- WdsPktStatisticsReq, 1063
  - pStatMask, 1063
- WdsPktStatisticsResp, 1063
  - pV4sessionId, 1064
  - pV6sessionId, 1064
  - PktStatElmntsV4, 1064
  - PktStatElmntsV6, 1064
- WdsProfileParam, 1064
  - SlqsProfile3GPP, 1064
  - SlqsProfile3GPP2, 1064
- WdsRunTimeSettings, 1064
  - rts, 1065
  - v4sessionId, 1065
  - v6sessionId, 1065
- wdsSetEventReportReq, 1065
  - pCurrChannelRateInd, 1067
  - pCurrDataBearerTechInd, 1067
  - pCurrPrefDataSysInd, 1067
  - pDataBearerTechInd, 1067
  - pDataCallStatusChangeInd, 1067
  - pDataSystemStatusChangeInd, 1067
  - pDormancyStatusInd, 1067
  - pEVDOPageMonPerChangeInd, 1067
  - pMIPStatusInd, 1067
  - pTransferStatInd, 1067
- Wireless Data Service (WDS), 32
- xAxis
  - sensorData, 676
- xferStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 970
- xid
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, 530

- pack\_qmi\_t, [549](#)
  - unpack\_qmi\_t, [918](#)
- xtra\_start\_gps\_minutes
  - GPSSStateInfo, [266](#)
- xtra\_start\_gps\_week
  - GPSSStateInfo, [266](#)
- xtra\_valid\_duration\_hours
  - GPSSStateInfo, [266](#)
- yAxis
  - sensorData, [676](#)
- year
  - nas\_timeInfo, [456](#)
  - nas\_UniversalTime, [461](#)
  - timeInfo, [784](#)
  - UniversalTime, [843](#)
- zAxis
  - sensorData, [676](#)