

LinuxQMI SDK-Lite  
SLQS04.00.24

Generated by Doxygen 1.8.6

Fri Feb 14 2020 02:35:32



# 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>29</b>
5.1	File List . . . . .	29
<b>6</b>	<b>Module Documentation</b>	<b>31</b>
6.1	QMI pack/unpack (lite-qmi) . . . . .	31
6.1.1	Detailed Description . . . . .	31
6.2	Streaming Download Protocol (lite-fw) . . . . .	32
6.2.1	Detailed Description . . . . .	32
<b>7</b>	<b>Namespace Documentation</b>	<b>33</b>
7.1	Tables Namespace Reference . . . . .	33
7.1.1	Detailed Description . . . . .	33
<b>8</b>	<b>Data Structure Documentation</b>	<b>35</b>
8.1	_litelfw_FirmwareFileInfo Struct Reference . . . . .	35
8.1.1	Detailed Description . . . . .	35
8.1.2	Field Documentation . . . . .	36

8.1.2.1	carrierStr	36
8.1.2.2	fullPath	36
8.1.2.3	headerType	36
8.1.2.4	imageMask	36
8.1.2.5	imageType	36
8.1.2.6	modelIdStr	36
8.1.2.7	packageIdStr	36
8.1.2.8	partNoStr	36
8.1.2.9	prVersionStr	36
8.1.2.10	releaseDate	36
8.1.2.11	skuStr	36
8.1.2.12	versionStr	36
8.2	_litefw_FirmwareInfo_ Struct Reference	36
8.2.1	Detailed Description	37
8.2.2	Field Documentation	37
8.2.2.1	szCarrier_str	37
8.2.2.2	szCarrierPrVersion_str	37
8.2.2.3	szFwversion_str	37
8.2.2.4	szModelid_str	37
8.2.2.5	szPackageid_str	37
8.2.2.6	szSku_str	37
8.3	_litefw_FirmwarePartNo_ Struct Reference	37
8.3.1	Detailed Description	38
8.3.2	Field Documentation	38
8.3.2.1	szPartno_str	38
8.4	altSrcInfo_t Struct Reference	38
8.4.1	Detailed Description	38
8.4.2	Field Documentation	39
8.4.2.1	coverage	39
8.4.2.2	linkage	39
8.4.2.3	source	39
8.5	appStats Struct Reference	39
8.5.1	Detailed Description	39
8.5.2	Field Documentation	41
8.5.2.1	aidLength	41
8.5.2.2	aidVal	41
8.5.2.3	appState	41
8.5.2.4	appType	41
8.5.2.5	persoFeature	41
8.5.2.6	persoRetries	41

8.5.2.7	<a href="#">persoState</a>	41
8.5.2.8	<a href="#">persoUnblockRetries</a>	42
8.5.2.9	<a href="#">pin1Retries</a>	42
8.5.2.10	<a href="#">pin1State</a>	42
8.5.2.11	<a href="#">pin2Retries</a>	42
8.5.2.12	<a href="#">pin2State</a>	42
8.5.2.13	<a href="#">puk1Retries</a>	42
8.5.2.14	<a href="#">puk2Retries</a>	42
8.5.2.15	<a href="#">univPin</a>	42
8.6	<a href="#">audio_RXAGCList Struct Reference</a>	42
8.6.1	<a href="#">Detailed Description</a>	42
8.6.2	<a href="#">Field Documentation</a>	42
8.6.2.1	<a href="#">pRXAIG</a>	43
8.6.2.2	<a href="#">pRXComprSlope</a>	43
8.6.2.3	<a href="#">pRXComprThres</a>	43
8.6.2.4	<a href="#">pRXExpSlope</a>	43
8.6.2.5	<a href="#">pRXExpThres</a>	43
8.6.2.6	<a href="#">pRXStaticGain</a>	43
8.7	<a href="#">audio_RXAVCList Struct Reference</a>	43
8.7.1	<a href="#">Detailed Description</a>	43
8.7.2	<a href="#">Field Documentation</a>	43
8.7.2.1	<a href="#">pAVRXAVCHedroom</a>	43
8.7.2.2	<a href="#">pAVRXAVCSens</a>	43
8.8	<a href="#">audio_RXPCMIIRFtr Struct Reference</a>	43
8.8.1	<a href="#">Detailed Description</a>	43
8.8.2	<a href="#">Field Documentation</a>	45
8.8.2.1	<a href="#">pFlag</a>	45
8.8.2.2	<a href="#">pStage0Val</a>	45
8.8.2.3	<a href="#">pStage1Val</a>	45
8.8.2.4	<a href="#">pStage2Val</a>	45
8.8.2.5	<a href="#">pStage3Val</a>	45
8.8.2.6	<a href="#">pStage4Val</a>	45
8.8.2.7	<a href="#">pStageCnt</a>	45
8.9	<a href="#">audio_TXAGCList Struct Reference</a>	45
8.9.1	<a href="#">Detailed Description</a>	45
8.9.2	<a href="#">Field Documentation</a>	45
8.9.2.1	<a href="#">pTXAIG</a>	46
8.9.2.2	<a href="#">pTXComprSlope</a>	46
8.9.2.3	<a href="#">pTXComprThres</a>	46
8.9.2.4	<a href="#">pTXExpSlope</a>	46

8.9.2.5	pTXExpThres	46
8.9.2.6	pTXStaticGain	46
8.10	audio_TXPCMIIRFiltr Struct Reference	46
8.10.1	Detailed Description	46
8.10.2	Field Documentation	47
8.10.2.1	pFlag	47
8.10.2.2	pStage0Val	47
8.10.2.3	pStage1Val	47
8.10.2.4	pStage2Val	47
8.10.2.5	pStage3Val	47
8.10.2.6	pStage4Val	47
8.10.2.7	pStageCnt	47
8.11	CarrierImage_t Struct Reference	47
8.11.1	Detailed Description	48
8.11.2	Field Documentation	48
8.11.2.1	m_FwBuildId	48
8.11.2.2	m_FwImageld	48
8.11.2.3	m_nCarrierId	48
8.11.2.4	m_nFolderId	48
8.11.2.5	m_nStorage	48
8.11.2.6	m_PriBuildId	49
8.11.2.7	m_PrImageld	49
8.12	cat_AlPhalIdentifierTlv Struct Reference	49
8.12.1	Detailed Description	49
8.12.2	Field Documentation	49
8.12.2.1	AlphaID	49
8.12.2.2	AlphaIDLength	49
8.12.2.3	ReferenceID	49
8.13	cat_commonEventTlv Struct Reference	49
8.13.1	Detailed Description	49
8.13.2	Field Documentation	50
8.13.2.1	CatEvent	50
8.13.2.2	EventID	50
8.13.2.3	EventLength	50
8.13.2.4	TlvPresent	50
8.14	cat_currentCatEvent Union Reference	50
8.14.1	Detailed Description	50
8.14.2	Field Documentation	50
8.14.2.1	CatAlphaldtfr	50
8.14.2.2	CatEndPS	50

8.14.2.3	CatEventLst	50
8.14.2.4	CatEvIDData	50
8.14.2.5	CatRefresh	50
8.15	cat_EndProactiveSessionTlv Struct Reference	51
8.15.1	Detailed Description	51
8.15.2	Field Documentation	51
8.15.2.1	EndProactiveSession	51
8.16	cat_EventIDDataTlv Struct Reference	51
8.16.1	Detailed Description	51
8.16.2	Field Documentation	51
8.16.2.1	Data	51
8.16.2.2	DataLength	51
8.16.2.3	ReferenceID	51
8.17	cat_EventListTlv Struct Reference	51
8.17.1	Detailed Description	52
8.17.2	Field Documentation	52
8.17.2.1	SetupEventList	52
8.18	cat_RefreshTlv Struct Reference	52
8.18.1	Detailed Description	52
8.18.2	Field Documentation	52
8.18.2.1	RefreshMode	52
8.18.2.2	RefreshStage	52
8.19	cdmaSSInfo Struct Reference	52
8.19.1	Detailed Description	53
8.19.2	Field Documentation	53
8.19.2.1	ecio	53
8.19.2.2	rsi	53
8.20	connectionStatus Struct Reference	53
8.20.1	Detailed Description	53
8.20.2	Field Documentation	53
8.20.2.1	MDMCallDuration	53
8.20.2.2	MDMConnStatus	54
8.21	crashInfoParams Struct Reference	54
8.21.1	Detailed Description	54
8.21.2	Field Documentation	54
8.21.2.1	crashInfo	54
8.21.2.2	crashStatus	54
8.22	crashInformation Struct Reference	54
8.22.1	Detailed Description	54
8.22.2	Field Documentation	55

8.22.2.1	crashData	55
8.22.2.2	crashId	55
8.22.2.3	crashString	55
8.22.2.4	crashStrlen	55
8.22.2.5	gcdumpString	55
8.22.2.6	gcdumpStrlen	55
8.22.2.7	numCrashes	55
8.23	currNetworkInfo Struct Reference	55
8.23.1	Detailed Description	55
8.23.2	Field Documentation	56
8.23.2.1	NetworkType	56
8.23.2.2	RATMask	56
8.23.2.3	SOMask	56
8.24	dms_ActivationStatusTlv Struct Reference	56
8.24.1	Detailed Description	56
8.24.2	Field Documentation	56
8.24.2.1	activationStatus	56
8.24.2.2	TlvPresent	56
8.25	dms_devCaps Struct Reference	56
8.25.1	Detailed Description	57
8.25.2	Field Documentation	57
8.25.2.1	DataServiceCapability	57
8.25.2.2	MaxRXChannelRate	57
8.25.2.3	MaxTXChannelRate	57
8.25.2.4	Radiolfaces	57
8.25.2.5	RadiolfacesSize	57
8.25.2.6	SimCapability	58
8.26	dms_devCurSubsCaps Struct Reference	58
8.26.1	Detailed Description	58
8.26.2	Field Documentation	58
8.26.2.1	CurSubsCapsLen	58
8.26.2.2	SubsCapList	58
8.27	dms_devMaxCfgListCaps Struct Reference	58
8.27.1	Detailed Description	58
8.27.2	Field Documentation	59
8.27.2.1	CurlIndex	59
8.27.2.2	DevCfgListLen	59
8.27.2.3	MaxActive	59
8.27.2.4	MaxSubs	59
8.27.2.5	SubsDevList	59



8.28 dms_devMaxSubsCaps Struct Reference	59
8.28.1 Detailed Description	59
8.28.2 Field Documentation	60
8.28.2.1 MaxSubsCapLen	60
8.28.2.2 MaxSubsList	60
8.29 dms_devMultiSimCaps Struct Reference	60
8.29.1 Detailed Description	60
8.29.2 Field Documentation	60
8.29.2.1 MaxSubs	60
8.29.2.2 SubsCfgList	60
8.29.2.3 SubsCfgListLen	60
8.30 dms_devMultiSimVoiceDataCaps Struct Reference	61
8.30.1 Detailed Description	61
8.30.2 Field Documentation	61
8.30.2.1 MaxActive	61
8.30.2.2 MaxSubs	61
8.31 dms_devSubsCfgList Struct Reference	61
8.31.1 Detailed Description	61
8.31.2 Field Documentation	62
8.31.2.1 MaxActive	62
8.31.2.2 SubsList	62
8.31.2.3 SubsListLen	62
8.32 dms_devSubsFeatureModeCaps Struct Reference	62
8.32.1 Detailed Description	62
8.32.2 Field Documentation	63
8.32.2.1 SubsFeatureLen	63
8.32.2.2 SubsFeatureList	63
8.33 dms_devSubsList Struct Reference	63
8.33.1 Detailed Description	63
8.33.2 Field Documentation	63
8.33.2.1 SubsList	63
8.33.2.2 SubsListLen	63
8.34 dms_devSubsVoiceDataCaps Struct Reference	63
8.34.1 Detailed Description	63
8.34.2 Field Documentation	64
8.34.2.1 SubsVoiceDataCapLen	64
8.34.2.2 SubsVoiceDataList	64
8.35 dms_devSubsVoiceDataList Struct Reference	64
8.35.1 Detailed Description	64
8.35.2 Field Documentation	64

8.35.2.1	SimVoiceDataCap	64
8.35.2.2	SubsVoiceDataCap	64
8.36	dms_LteBandsSupport Struct Reference	64
8.36.1	Detailed Description	65
8.36.2	Field Documentation	65
8.36.2.1	lteBands	65
8.36.2.2	supportedLteBandLen	65
8.36.2.3	TLVPresent	65
8.37	dms_OperatingModeTlv Struct Reference	65
8.37.1	Detailed Description	65
8.37.2	Field Documentation	66
8.37.2.1	operatingMode	66
8.37.2.2	TlvPresent	66
8.38	dms_PSMActiveTimerIndTlv Struct Reference	66
8.38.1	Detailed Description	66
8.38.2	Field Documentation	66
8.38.2.1	ActiveTimerInd	66
8.38.2.2	TlvPresent	66
8.39	dms_PSMActiveTimerTlv Struct Reference	66
8.39.1	Detailed Description	67
8.39.2	Field Documentation	67
8.39.2.1	activeTimer	67
8.39.2.2	TlvPresent	67
8.40	dms_PSMDurationDueToOOSTlv Struct Reference	67
8.40.1	Detailed Description	67
8.40.2	Field Documentation	67
8.40.2.1	durationDueToOOS	67
8.40.2.2	TlvPresent	67
8.41	dms_PSMDurationThresholdTlv Struct Reference	67
8.41.1	Detailed Description	68
8.41.2	Field Documentation	68
8.41.2.1	durationThreshold	68
8.41.2.2	TlvPresent	68
8.42	dms_PSMEarlyWakeupTimeTlv Struct Reference	68
8.42.1	Detailed Description	68
8.42.2	Field Documentation	68
8.42.2.1	earlyWakeupTime	68
8.42.2.2	TlvPresent	68
8.43	dms_PSMEnableStateIndTlv Struct Reference	69
8.43.1	Detailed Description	69

8.43.2	Field Documentation	69
8.43.2.1	EnableStateInd	69
8.43.2.2	TlvPresent	69
8.44	dms_PSMEnableStateTlv Struct Reference	69
8.44.1	Detailed Description	69
8.44.2	Field Documentation	69
8.44.2.1	enableState	70
8.44.2.2	TlvPresent	70
8.45	dms_PSMPeriodicUpdateTimerIndTlv Struct Reference	70
8.45.1	Detailed Description	70
8.45.2	Field Documentation	70
8.45.2.1	PeriodicUpdateTimerInd	70
8.45.2.2	TlvPresent	70
8.46	dms_PSMPeriodicUpdateTimerTlv Struct Reference	70
8.46.1	Detailed Description	70
8.46.2	Field Documentation	71
8.46.2.1	periodicUpdateTimer	71
8.46.2.2	TlvPresent	71
8.47	dms_PSMRandomizationWindowTlv Struct Reference	71
8.47.1	Detailed Description	71
8.47.2	Field Documentation	71
8.47.2.1	randomizationWindow	71
8.47.2.2	TlvPresent	71
8.48	dms_TemperatureTlv Struct Reference	71
8.48.1	Detailed Description	71
8.48.2	Field Documentation	72
8.48.2.1	Temperature	72
8.48.2.2	TempStat	72
8.48.2.3	TlvPresent	72
8.49	dms_UimAutoSwitchActSlotTlv Struct Reference	72
8.49.1	Detailed Description	72
8.49.2	Field Documentation	72
8.49.2.1	TlvPresent	72
8.49.2.2	uimAutoSwitchActSlot	73
8.50	dms_UimStatusTlv Struct Reference	73
8.50.1	Detailed Description	73
8.50.2	Field Documentation	73
8.50.2.1	event	73
8.50.2.2	intf	73
8.50.2.3	TlvPresent	73

8.51	dms_VoltageTlv Struct Reference	73
8.51.1	Detailed Description	73
8.51.2	Field Documentation	74
8.51.2.1	TlvPresent	74
8.51.2.2	Voltage	74
8.51.2.3	VoltStat	74
8.52	DMScustSettingInfo Struct Reference	74
8.52.1	Detailed Description	74
8.52.2	Field Documentation	75
8.52.2.1	cust_attr	75
8.52.2.2	cust_id	75
8.52.2.3	cust_value	75
8.52.2.4	id_length	75
8.52.2.5	value_length	75
8.53	DMScustSettingList Struct Reference	75
8.53.1	Detailed Description	75
8.53.2	Field Documentation	75
8.53.2.1	custSetting	75
8.53.2.2	list_type	75
8.53.2.3	num_instances	75
8.54	DMSgetCustomFeatureV2 Struct Reference	76
8.54.1	Detailed Description	76
8.54.2	Field Documentation	76
8.54.2.1	pCustSettingInfo	76
8.54.2.2	pCustSettingList	76
8.54.2.3	pGetCustomInput	76
8.55	DMSgetCustomInput Struct Reference	76
8.55.1	Detailed Description	76
8.55.2	Field Documentation	77
8.55.2.1	cust_id	77
8.55.2.2	list_type	77
8.56	dunchannelRate Struct Reference	77
8.56.1	Detailed Description	77
8.56.2	Field Documentation	77
8.56.2.1	CurrChanRxRate	77
8.56.2.2	CurrChanTxRate	77
8.56.2.3	MaxChanRxRate	77
8.56.2.4	MaxChanTxRate	77
8.57	eriDataparams Struct Reference	77
8.57.1	Detailed Description	78

8.57.2	Field Documentation	78
8.57.2.1	eriData	78
8.57.2.2	eriDataLen	78
8.58	eTWSPLMNInfoTlv Struct Reference	78
8.58.1	Detailed Description	78
8.58.2	Field Documentation	78
8.58.2.1	ETWSPLMNInfo	78
8.58.2.2	TlvPresent	78
8.59	FMSImageElement Struct Reference	78
8.59.1	Detailed Description	79
8.59.2	Field Documentation	79
8.59.2.1	buildId	79
8.59.2.2	buildIdLength	79
8.59.2.3	imageId	79
8.59.2.4	imageType	79
8.60	FMSImageIdElement Struct Reference	79
8.60.1	Detailed Description	79
8.60.2	Field Documentation	80
8.60.2.1	buildID	80
8.60.2.2	buildIDLength	80
8.60.2.3	failureCount	80
8.60.2.4	imageID	80
8.60.2.5	storageIndex	80
8.61	FMSImageIDEntries Struct Reference	80
8.61.1	Detailed Description	80
8.61.2	Field Documentation	81
8.61.2.1	executingImage	81
8.61.2.2	imageIDElement	81
8.61.2.3	imageIDSize	81
8.61.2.4	imageType	81
8.61.2.5	maxImages	81
8.62	FMSImageList Struct Reference	81
8.62.1	Detailed Description	81
8.62.2	Field Documentation	81
8.62.2.1	imageIDEntries	81
8.62.2.2	listSize	81
8.63	FMSPrefImageList Struct Reference	82
8.63.1	Detailed Description	82
8.63.2	Field Documentation	82
8.63.2.1	listEntries	82

8.63.2.2	listSize	82
8.64	hdrSSInfo Struct Reference	82
8.64.1	Detailed Description	82
8.64.2	Field Documentation	83
8.64.2.1	ecio	83
8.64.2.2	io	83
8.64.2.3	rsi	83
8.64.2.4	sinr	83
8.65	image_info_t Struct Reference	83
8.65.1	Detailed Description	83
8.65.2	Field Documentation	84
8.65.2.1	buildID	84
8.65.2.2	buildIDLen	84
8.65.2.3	imageType	84
8.65.2.4	uniqueID	84
8.66	ims_AMRModelInfo Struct Reference	84
8.66.1	Detailed Description	84
8.66.2	Field Documentation	84
8.66.2.1	amrMode	85
8.66.2.2	TlvPresent	85
8.67	ims_AMROctAlgnInfo Struct Reference	85
8.67.1	Detailed Description	85
8.67.2	Field Documentation	85
8.67.2.1	amrOctAlgn	85
8.67.2.2	TlvPresent	85
8.68	ims_AMRWBModelInfo Struct Reference	85
8.68.1	Detailed Description	85
8.68.2	Field Documentation	86
8.68.2.1	amrWBMode	86
8.68.2.2	TlvPresent	86
8.69	ims_AMRWBOctAlgnInfo Struct Reference	86
8.69.1	Detailed Description	86
8.69.2	Field Documentation	86
8.69.2.1	amrWBOctAlgn	86
8.69.2.2	TlvPresent	86
8.70	ims_CSCFPortNameInfo Struct Reference	86
8.70.1	Detailed Description	87
8.70.2	Field Documentation	87
8.70.2.1	cscfPortName	87
8.70.2.2	TlvPresent	87

8.71	ims_EnabAMRWBInfo Struct Reference	87
8.71.1	Detailed Description	87
8.71.2	Field Documentation	87
8.71.2.1	amrWBEnable	87
8.71.2.2	TlvPresent	87
8.72	ims_EnabSCRAMRInfo Struct Reference	87
8.72.1	Detailed Description	87
8.72.2	Field Documentation	88
8.72.2.1	scrAmrEnable	88
8.72.2.2	TlvPresent	88
8.73	ims_EnabSCRAMRWBInfo Struct Reference	88
8.73.1	Detailed Description	88
8.73.2	Field Documentation	88
8.73.2.1	scrAmrWBEnable	88
8.73.2.2	TlvPresent	88
8.74	ims_IMSDomainInfo Struct Reference	88
8.74.1	Detailed Description	88
8.74.2	Field Documentation	89
8.74.2.1	imsDomainName	89
8.74.2.2	TlvPresent	89
8.75	ims_IMSTestModelInfo Struct Reference	89
8.75.1	Detailed Description	89
8.75.2	Field Documentation	89
8.75.2.1	imsTestMode	89
8.75.2.2	TlvPresent	89
8.76	ims_MinSessExpInfo Struct Reference	89
8.76.1	Detailed Description	89
8.76.2	Field Documentation	90
8.76.2.1	minSessExp	90
8.76.2.2	TlvPresent	90
8.77	ims_PCSCFPortInfo Struct Reference	90
8.77.1	Detailed Description	90
8.77.2	Field Documentation	90
8.77.2.1	priCSCFPort	90
8.77.2.2	TlvPresent	90
8.78	ims_PhCtxtURIInfo Struct Reference	90
8.78.1	Detailed Description	90
8.78.2	Field Documentation	90
8.78.2.1	PhCtxtURI	91
8.78.2.2	TlvPresent	91

8.79	ims_RngBkTmrInfo Struct Reference	91
8.79.1	Detailed Description	91
8.79.2	Field Documentation	91
8.79.2.1	RingBkTmr	91
8.79.2.2	TlvPresent	91
8.80	ims_RngTmrInfo Struct Reference	91
8.80.1	Detailed Description	91
8.80.2	Field Documentation	91
8.80.2.1	RingTmr	92
8.80.2.2	TlvPresent	92
8.81	ims_RTPRTCPInactTmrDurInfo Struct Reference	92
8.81.1	Detailed Description	92
8.81.2	Field Documentation	92
8.81.2.1	InactTmr	92
8.81.2.2	TlvPresent	92
8.82	ims_SessDurInfo Struct Reference	92
8.82.1	Detailed Description	92
8.82.2	Field Documentation	92
8.82.2.1	sessExp	93
8.82.2.2	TlvPresent	93
8.83	ims_SigCompEnInfo Struct Reference	93
8.83.1	Detailed Description	93
8.83.2	Field Documentation	93
8.83.2.1	SigCompEn	93
8.83.2.2	TlvPresent	93
8.84	ims_SIPPortInfo Struct Reference	93
8.84.1	Detailed Description	93
8.84.2	Field Documentation	93
8.84.2.1	SIPLocalPort	93
8.84.2.2	TlvPresent	93
8.85	ims_SIPRegnTmrInfo Struct Reference	94
8.85.1	Detailed Description	94
8.85.2	Field Documentation	94
8.85.2.1	TlvPresent	94
8.85.2.2	tmrSIPRegn	94
8.86	ims_SMSFmtInfo Struct Reference	94
8.86.1	Detailed Description	94
8.86.2	Field Documentation	94
8.86.2.1	smsFormat	94
8.86.2.2	TlvPresent	94



8.87	ims_SMSolPNwInfo Struct Reference	95
8.87.1	Detailed Description	95
8.87.2	Field Documentation	95
8.87.2.1	smsolPNW	95
8.87.2.2	TlvPresent	95
8.88	ims_SubscrTmrInfo Struct Reference	95
8.88.1	Detailed Description	95
8.88.2	Field Documentation	95
8.88.2.1	subscrTmr	95
8.88.2.2	TlvPresent	95
8.89	ims_TmrT1Info Struct Reference	95
8.89.1	Detailed Description	96
8.89.2	Field Documentation	96
8.89.2.1	TlvPresent	96
8.89.2.2	tmrT1	96
8.90	ims_TmrT2Info Struct Reference	96
8.90.1	Detailed Description	96
8.90.2	Field Documentation	96
8.90.2.1	TlvPresent	96
8.90.2.2	tmrT2	96
8.91	ims_TmrTfInfo Struct Reference	96
8.91.1	Detailed Description	97
8.91.2	Field Documentation	97
8.91.2.1	TlvPresent	97
8.91.2.2	tmrTf	97
8.92	imsa_IMSFailErrCodeTlv Struct Reference	97
8.92.1	Detailed Description	97
8.92.2	Field Documentation	97
8.92.2.1	ImsFailErrCode	97
8.92.2.2	TlvPresent	97
8.93	imsa_IMSRegStatusErrorCodeInfo Struct Reference	97
8.93.1	Detailed Description	98
8.93.2	Field Documentation	98
8.93.2.1	ErrorCode	98
8.93.2.2	TlvPresent	98
8.94	imsa_IMSRegStatusInfo Struct Reference	98
8.94.1	Detailed Description	98
8.94.2	Field Documentation	98
8.94.2.1	ImsRegistered	98
8.94.2.2	TlvPresent	98

8.95	imsa_NewIMSRegStatusInfo Struct Reference	99
8.95.1	Detailed Description	99
8.95.2	Field Documentation	99
8.95.2.1	ImsRegStatus	99
8.95.2.2	TlvPresent	99
8.96	imsa_RatHandoverStatusInfo Struct Reference	99
8.96.1	Detailed Description	99
8.96.2	Field Documentation	100
8.96.2.1	ErrorCodeData	100
8.96.2.2	ErrorCodeLen	100
8.96.2.3	RatHandoverStatus	100
8.96.2.4	SourceRAT	100
8.96.2.5	TargetRAT	100
8.96.2.6	TlvPresent	100
8.97	imsa_SmsRatInfo Struct Reference	100
8.97.1	Detailed Description	100
8.97.2	Field Documentation	100
8.97.2.1	SmsRatVal	101
8.97.2.2	TlvPresent	101
8.98	imsa_SmsSvcStatusInfo Struct Reference	101
8.98.1	Detailed Description	101
8.98.2	Field Documentation	101
8.98.2.1	SmsSvcStatus	101
8.98.2.2	TlvPresent	101
8.99	imsa_UtRatInfo Struct Reference	101
8.99.1	Detailed Description	101
8.99.2	Field Documentation	102
8.99.2.1	TlvPresent	102
8.99.2.2	UtRatVal	102
8.100	imsa_UtSvcStatusInfo Struct Reference	102
8.100.1	Detailed Description	102
8.100.2	Field Documentation	102
8.100.2.1	TlvPresent	102
8.100.2.2	UtSvcStatus	102
8.101	imsa_VoipRatInfo Struct Reference	102
8.101.1	Detailed Description	102
8.101.2	Field Documentation	103
8.101.2.1	TlvPresent	103
8.101.2.2	VoipRatVal	103
8.102	imsa_VoipSvcStatusInfo Struct Reference	103

8.102.1 Detailed Description . . . . .	103
8.102.2 Field Documentation . . . . .	103
8.102.2.1 TlvPresent . . . . .	103
8.102.2.2 VoipSvcStatus . . . . .	103
8.103imsa_VtRatInfo Struct Reference . . . . .	103
8.103.1 Detailed Description . . . . .	103
8.103.2 Field Documentation . . . . .	104
8.103.2.1 TlvPresent . . . . .	104
8.103.2.2 VtRatVal . . . . .	104
8.104imsa_VtSvcStatusInfo Struct Reference . . . . .	104
8.104.1 Detailed Description . . . . .	104
8.104.2 Field Documentation . . . . .	104
8.104.2.1 TlvPresent . . . . .	104
8.104.2.2 VtSvcStatus . . . . .	104
8.105ipv6AddressInfo Struct Reference . . . . .	104
8.105.1 Detailed Description . . . . .	104
8.105.2 Field Documentation . . . . .	104
8.105.2.1 IPAddressV6 . . . . .	105
8.105.2.2 IPV6PrefixLen . . . . .	105
8.106LibPackGPRSRequestedQoS Struct Reference . . . . .	105
8.106.1 Detailed Description . . . . .	105
8.106.2 Field Documentation . . . . .	105
8.106.2.1 delayClass . . . . .	105
8.106.2.2 meanThroughputClass . . . . .	105
8.106.2.3 peakThroughputClass . . . . .	105
8.106.2.4 precedenceClass . . . . .	105
8.106.2.5 reliabilityClass . . . . .	105
8.107LibPackPCOIDList Struct Reference . . . . .	105
8.107.1 Detailed Description . . . . .	106
8.107.2 Field Documentation . . . . .	106
8.107.2.1 PcoList . . . . .	106
8.108LibPackPDNThrottleTimer Struct Reference . . . . .	106
8.108.1 Detailed Description . . . . .	106
8.108.2 Field Documentation . . . . .	106
8.108.2.1 ThrottleTimer . . . . .	106
8.109LibpackProfile3GPP Struct Reference . . . . .	106
8.109.1 Detailed Description . . . . .	107
8.109.2 Field Documentation . . . . .	112
8.109.2.1 pAddrAllocPref . . . . .	112
8.109.2.2 pAPNClass . . . . .	112

8.109.2.3 pAPNDisabledFlag . . . . .	112
8.109.2.4 pAPNName . . . . .	112
8.109.2.5 pAPNnameSize . . . . .	112
8.109.2.6 pAuthenticationPref . . . . .	112
8.109.2.7 pGPRSMinimumQoS . . . . .	112
8.109.2.8 pGPRSRequestedQos . . . . .	112
8.109.2.9 plmCnFlag . . . . .	112
8.109.2.10pIPv4AddrPref . . . . .	112
8.109.2.11pIPv6AddPref . . . . .	112
8.109.2.12pPassword . . . . .	112
8.109.2.13pPasswordSize . . . . .	112
8.109.2.14pPcsfAddrUsingDhcp . . . . .	112
8.109.2.15pPcsfAddrUsingPCO . . . . .	112
8.109.2.16pPDNInactivTimeout . . . . .	112
8.109.2.17pPdpAccessConFlag . . . . .	112
8.109.2.18pPdpContext . . . . .	112
8.109.2.19pPdpDataCompType . . . . .	112
8.109.2.20pPdpHdrCompType . . . . .	112
8.109.2.21pPDType . . . . .	112
8.109.2.22pPriDNSIPv4AddPref . . . . .	112
8.109.2.23pPriDNSIPv6addpref . . . . .	112
8.109.2.24pPrimaryID . . . . .	112
8.109.2.25pProfilename . . . . .	112
8.109.2.26pProfilenameSize . . . . .	112
8.109.2.27pQosClassID . . . . .	112
8.109.2.28pSecDNSIPv4AddPref . . . . .	113
8.109.2.29pSecDNSIPv6addpref . . . . .	113
8.109.2.30pSecondaryFlag . . . . .	113
8.109.2.31pSupportEmergencyCalls . . . . .	113
8.109.2.32pTFTID1Params . . . . .	113
8.109.2.33pTFTID2Params . . . . .	113
8.109.2.34pUMTSMinQoS . . . . .	113
8.109.2.35pUMTSMinQoSSigInd . . . . .	113
8.109.2.36pUMTSReqQoS . . . . .	113
8.109.2.37pUMTSReqQoSSigInd . . . . .	113
8.109.2.38pUsername . . . . .	113
8.109.2.39pUsernameSize . . . . .	113
8.110LibpackProfile3GPP2 Struct Reference . . . . .	113
8.110.1 Detailed Description . . . . .	114
8.110.2 Field Documentation . . . . .	118

8.110.2.1 pAllowLinger . . . . .	118
8.110.2.2 pAPNClass3GPP2 . . . . .	118
8.110.2.3 pAPNEnabled3GPP2 . . . . .	118
8.110.2.4 pApnString . . . . .	118
8.110.2.5 pApnStringSize . . . . .	118
8.110.2.6 pAppPriority . . . . .	118
8.110.2.7 pAppType . . . . .	118
8.110.2.8 pAuthPassword . . . . .	118
8.110.2.9 pAuthPasswordSize . . . . .	118
8.110.2.10 pAuthProtocol . . . . .	118
8.110.2.11 pAuthRetryCount . . . . .	118
8.110.2.12 pAuthTimeout . . . . .	118
8.110.2.13 pDataMode . . . . .	118
8.110.2.14 pDataRate . . . . .	118
8.110.2.15 pIpcpAckTimeout . . . . .	118
8.110.2.16 pIpcpCreqRetryCount . . . . .	118
8.110.2.17 pIsPcscfAddressNedded . . . . .	118
8.110.2.18 pLcpAckTimeout . . . . .	118
8.110.2.19 pLcpCreqRetryCount . . . . .	118
8.110.2.20 pNegoDnsSrvrPref . . . . .	118
8.110.2.21 pPDNInactivTimeout3GPP2 . . . . .	118
8.110.2.22 pPdnType . . . . .	118
8.110.2.23 pPppSessCloseTimer1x . . . . .	119
8.110.2.24 pPppSessCloseTimerDO . . . . .	119
8.110.2.25 pPrimaryV4DnsAddress . . . . .	119
8.110.2.26 pPriV6DnsAddress . . . . .	119
8.110.2.27 pRATType . . . . .	119
8.110.2.28 pSecondaryV4DnsAddress . . . . .	119
8.110.2.29 pSecV6DnsAddress . . . . .	119
8.110.2.30 pUserId . . . . .	119
8.110.2.31 pUserIdSize . . . . .	119
8.111 LibpackProfile3GPPV2 Struct Reference . . . . .	119
8.111.1 Detailed Description . . . . .	120
8.111.2 Field Documentation . . . . .	127
8.111.2.1 pAddrAllocPref . . . . .	127
8.111.2.2 pAPNBearer . . . . .	127
8.111.2.3 pAPNClass . . . . .	127
8.111.2.4 pAPNDisabledFlag . . . . .	127
8.111.2.5 pAPNName . . . . .	127
8.111.2.6 pAPNnameSize . . . . .	127

8.111.2.7 pAppUserData . . . . .	127
8.111.2.8 pAuthenticationPref . . . . .	127
8.111.2.9 pClatFlag . . . . .	127
8.111.2.10 pDnsWithDHCPFlag . . . . .	127
8.111.2.11 pGPRSMinimumQoS . . . . .	127
8.111.2.12 pGPRSRequestedQos . . . . .	128
8.111.2.13 pImCnFlag . . . . .	128
8.111.2.14 pIPv4AddrPref . . . . .	128
8.111.2.15 pIPv6AddrPref . . . . .	128
8.111.2.16 pIPv6DelegFlag . . . . .	128
8.111.2.17 pIWLANtoLTEHandoverFlag . . . . .	128
8.111.2.18 pLteRoamPDType . . . . .	128
8.111.2.19 pLTEtoIWLANHandoverFlag . . . . .	128
8.111.2.20 pMaxPDN . . . . .	128
8.111.2.21 pMaxPDNTimer . . . . .	128
8.111.2.22 pMcc . . . . .	128
8.111.2.23 pMnc . . . . .	128
8.111.2.24 pMsisdnFlag . . . . .	128
8.111.2.25 pOperatorPCOID . . . . .	128
8.111.2.26 pOverridePDType . . . . .	128
8.111.2.27 pPassword . . . . .	128
8.111.2.28 pPasswordSize . . . . .	128
8.111.2.29 pPCOIDList . . . . .	128
8.111.2.30 pPscfAddrUsingDhcp . . . . .	128
8.111.2.31 pPscfAddrUsingPCO . . . . .	128
8.111.2.32 pPDNDisconnectWaitTimer . . . . .	128
8.111.2.33 pPDNInactivTimeout . . . . .	128
8.111.2.34 pPDNThrottleTimer . . . . .	128
8.111.2.35 pPDNWaitTimer . . . . .	128
8.111.2.36 pPdpAccessConFlag . . . . .	128
8.111.2.37 pPdpContext . . . . .	128
8.111.2.38 pPdpDataCompType . . . . .	128
8.111.2.39 pPdpHdrCompType . . . . .	128
8.111.2.40 pPDType . . . . .	129
8.111.2.41 pPersistFlag . . . . .	129
8.111.2.42 pPriDNSIPv4AddPref . . . . .	129
8.111.2.43 pPriDNSIPv6addpref . . . . .	129
8.111.2.44 pPrimaryID . . . . .	129
8.111.2.45 pProfilename . . . . .	129
8.111.2.46 pProfilenameSize . . . . .	129

8.111.2.47pQosClassID . . . . .	129
8.111.2.48pRoamDisallowFlag . . . . .	129
8.111.2.49pSecDNSIPv4AddPref . . . . .	129
8.111.2.50pSecDNSIPv6addpref . . . . .	129
8.111.2.51pSecondaryFlag . . . . .	129
8.111.2.52pSupportEmergencyCalls . . . . .	129
8.111.2.53pTFTID1Params . . . . .	129
8.111.2.54pTFTID2Params . . . . .	129
8.111.2.55pUMTSMinQoS . . . . .	129
8.111.2.56pUMTSMinQoSInd . . . . .	129
8.111.2.57pUMTSReqQoS . . . . .	129
8.111.2.58pUMTSReqQoSSigInd . . . . .	129
8.111.2.59pUmtsRoamPDPTYPE . . . . .	129
8.111.2.60pUsername . . . . .	129
8.111.2.61pUsernameSize . . . . .	129
8.112LibPackprofile_3GPP Struct Reference . . . . .	129
8.112.1 Detailed Description . . . . .	130
8.112.2 Field Documentation . . . . .	134
8.112.2.1 pAddrAllocPref . . . . .	134
8.112.2.2 pAPNClass . . . . .	134
8.112.2.3 pAPNDisabledFlag . . . . .	134
8.112.2.4 pAPNName . . . . .	134
8.112.2.5 pAPNnameSize . . . . .	134
8.112.2.6 pAuthenticationPref . . . . .	134
8.112.2.7 pGPRSMinimumQoS . . . . .	134
8.112.2.8 pGPRSRequestedQoS . . . . .	134
8.112.2.9 plmCnFlag . . . . .	134
8.112.2.10pIPv4AddrPref . . . . .	134
8.112.2.11pIPv6AddPref . . . . .	134
8.112.2.12pPassword . . . . .	134
8.112.2.13pPasswordSize . . . . .	134
8.112.2.14pPcscfAddrUsingDhcp . . . . .	134
8.112.2.15pPcscfAddrUsingPCO . . . . .	134
8.112.2.16pPDNInactivTimeout . . . . .	134
8.112.2.17pPdpAccessConFlag . . . . .	134
8.112.2.18pPdpContext . . . . .	134
8.112.2.19pPdpDataCompType . . . . .	134
8.112.2.20pPdpHdrCompType . . . . .	135
8.112.2.21pPDPTYPE . . . . .	135
8.112.2.22pPriDNSIPv4AddPref . . . . .	135

8.112.2.23	pPriDNSIPv6addpref	135
8.112.2.24	pPrimaryID	135
8.112.2.25	pProfilename	135
8.112.2.26	pProfilenameSize	135
8.112.2.27	pQosClassID	135
8.112.2.28	pSecDNSIPv4AddPref	135
8.112.2.29	pSecDNSIPv6addpref	135
8.112.2.30	pSecondaryFlag	135
8.112.2.31	pSupportEmergencyCalls	135
8.112.2.32	pTFTID1Params	135
8.112.2.33	pTFTID2Params	135
8.112.2.34	pUMTSMinQoS	135
8.112.2.35	pUMTSMinQoSSigInd	135
8.112.2.36	pUMTSReqQoS	135
8.112.2.37	pUMTSReqQoSSigInd	135
8.112.2.38	pUsername	135
8.112.2.39	pUsernameSize	135
8.113	LibPackprofile_3GPP2 Struct Reference	135
8.113.1	Detailed Description	136
8.113.2	Field Documentation	139
8.113.2.1	pAllowLinger	139
8.113.2.2	pAPNClass3GPP2	139
8.113.2.3	pAPNEnabled3GPP2	139
8.113.2.4	pApnString	139
8.113.2.5	pApnStringSize	140
8.113.2.6	pAppPriority	140
8.113.2.7	pAppType	140
8.113.2.8	pAuthPassword	140
8.113.2.9	pAuthPassword_tSize	140
8.113.2.10	pAuthProtocol	140
8.113.2.11	pAuthRetryCount	140
8.113.2.12	pAuthTimeout	140
8.113.2.13	pDataMode	140
8.113.2.14	pDataRate	140
8.113.2.15	pIpcpAckTimeout	140
8.113.2.16	pIpcpCreqRetryCount	140
8.113.2.17	pIsPcscfAddressNedded	140
8.113.2.18	pLcpAckTimeout	140
8.113.2.19	pLcpCreqRetryCount	140
8.113.2.20	pNegoDnsSrvrPref	140



8.113.2.21	pPDNInactivTimeout3GPP2	140
8.113.2.22	pPdnType	140
8.113.2.23	pPppSessCloseTimer1x	140
8.113.2.24	pPppSessCloseTimerDO	140
8.113.2.25	pPrimaryV4DnsAddress	140
8.113.2.26	pPriV6DnsAddress	140
8.113.2.27	pRATType	140
8.113.2.28	pSecondaryV4DnsAddress	140
8.113.2.29	pSecV6DnsAddress	140
8.113.2.30	pUserId	140
8.113.2.31	pUserIdSize	140
8.114	LibPackProfileMnc Struct Reference	140
8.114.1	Detailed Description	141
8.114.2	Field Documentation	141
8.114.2.1	MNC	141
8.114.2.2	PCSFlag	141
8.115	LibPackQosClassID Struct Reference	141
8.115.1	Detailed Description	141
8.115.2	Field Documentation	142
8.115.2.1	gDIBitRate	142
8.115.2.2	gUIBitRate	142
8.115.2.3	maxDIBitRate	142
8.115.2.4	maxUIBitRate	142
8.115.2.5	QCI	142
8.116	LibPackTFTIDParams Struct Reference	142
8.116.1	Detailed Description	142
8.116.2	Field Documentation	143
8.116.2.1	destPortRangeEnd	143
8.116.2.2	destPortRangeStart	143
8.116.2.3	eValid	143
8.116.2.4	filterId	143
8.116.2.5	flowLabel	143
8.116.2.6	IPSECSPi	144
8.116.2.7	ipVersion	144
8.116.2.8	nextHeader	144
8.116.2.9	pSourceIP	144
8.116.2.10	sourceIPMask	144
8.116.2.11	srcPortRangeEnd	144
8.116.2.12	srcPortRangeStart	144
8.116.2.13	tosMask	144

8.117LibPackUMTSQoS Struct Reference . . . . .	144
8.117.1 Detailed Description . . . . .	144
8.117.2 Field Documentation . . . . .	146
8.117.2.1 deliveryErrSDU . . . . .	146
8.117.2.2 grntDownlinkBitrate . . . . .	146
8.117.2.3 grntUplinkBitrate . . . . .	146
8.117.2.4 maxDownlinkBitrate . . . . .	146
8.117.2.5 maxSDUSize . . . . .	146
8.117.2.6 maxUplinkBitrate . . . . .	146
8.117.2.7 qosDeliveryOrder . . . . .	146
8.117.2.8 resBerRatio . . . . .	146
8.117.2.9 sduErrorRatio . . . . .	146
8.117.2.10trafficClass . . . . .	146
8.117.2.11trafficPriority . . . . .	146
8.117.2.12transferDelay . . . . .	146
8.118LibPackUMTSReqQoSsigInd Struct Reference . . . . .	147
8.118.1 Detailed Description . . . . .	147
8.118.2 Field Documentation . . . . .	147
8.118.2.1 SigInd . . . . .	147
8.118.2.2 UMTSReqQoS . . . . .	147
8.119loc_accelAcceptReady Struct Reference . . . . .	147
8.119.1 Detailed Description . . . . .	147
8.119.2 Field Documentation . . . . .	148
8.119.2.1 batchPerSec . . . . .	148
8.119.2.2 injectEnable . . . . .	148
8.119.2.3 samplesPerBatch . . . . .	148
8.120loc_accelTempAcceptReady Struct Reference . . . . .	148
8.120.1 Detailed Description . . . . .	148
8.120.2 Field Documentation . . . . .	149
8.120.2.1 batchPerSec . . . . .	149
8.120.2.2 injectEnable . . . . .	149
8.120.2.3 samplesPerBatch . . . . .	149
8.121loc_AppProviderInfoTlv Struct Reference . . . . .	149
8.121.1 Detailed Description . . . . .	149
8.121.2 Field Documentation . . . . .	150
8.121.2.1 name . . . . .	150
8.121.2.2 nameLen . . . . .	150
8.121.2.3 provider . . . . .	150
8.121.2.4 providerLen . . . . .	150
8.121.2.5 TlvPresent . . . . .	150

8.121.2.6 version	150
8.121.2.7 versionLen	150
8.121.2.8 verValid	150
8.122loc_BdsSV Struct Reference	150
8.122.1 Detailed Description	150
8.122.2 Field Documentation	150
8.122.2.1 id	150
8.122.2.2 mask	150
8.123loc_BdsSVInfo Struct Reference	151
8.123.1 Detailed Description	151
8.123.2 Field Documentation	151
8.123.2.1 len	151
8.123.2.2 pSV	151
8.124loc_CellDb Struct Reference	151
8.124.1 Detailed Description	151
8.124.2 Field Documentation	152
8.124.2.1 mask	152
8.125loc_ClkInfo Struct Reference	152
8.125.1 Detailed Description	152
8.125.2 Field Documentation	153
8.125.2.1 mask	153
8.126loc_FixCriteriaStatusTlv Struct Reference	153
8.126.1 Detailed Description	154
8.126.2 Field Documentation	154
8.126.2.1 status	154
8.126.2.2 TlvPresent	154
8.127loc_GnssData Struct Reference	154
8.127.1 Detailed Description	154
8.127.2 Field Documentation	155
8.127.2.1 mask	155
8.128loc_gpsTime Struct Reference	155
8.128.1 Detailed Description	156
8.128.2 Field Documentation	156
8.128.2.1 gpsTimeOfWeekMs	156
8.128.2.2 gpsWeek	156
8.129loc_gyroAcceptReady Struct Reference	156
8.129.1 Detailed Description	156
8.129.2 Field Documentation	157
8.129.2.1 batchPerSec	157
8.129.2.2 injectEnable	157

8.129.2.3 samplesPerBatch . . . . .	157
8.130loc_gyroTempAcceptReady Struct Reference . . . . .	157
8.130.1 Detailed Description . . . . .	157
8.130.2 Field Documentation . . . . .	157
8.130.2.1 batchPerSec . . . . .	157
8.130.2.2 injectEnable . . . . .	157
8.130.2.3 samplesPerBatch . . . . .	158
8.131loc_HorAccuracyLvlTlv Struct Reference . . . . .	158
8.131.1 Detailed Description . . . . .	158
8.131.2 Field Documentation . . . . .	158
8.131.2.1 accuracy . . . . .	158
8.131.2.2 TlvPresent . . . . .	158
8.132loc_IntermediateRptStateTlv Struct Reference . . . . .	158
8.132.1 Detailed Description . . . . .	158
8.132.2 Field Documentation . . . . .	159
8.132.2.1 intermediate . . . . .	159
8.132.2.2 TlvPresent . . . . .	159
8.133loc_IPv4Config Struct Reference . . . . .	159
8.133.1 Detailed Description . . . . .	159
8.133.2 Field Documentation . . . . .	159
8.133.2.1 IPv4Addr . . . . .	159
8.133.2.2 IPv4Port . . . . .	159
8.134loc_IPv4Info Struct Reference . . . . .	159
8.134.1 Detailed Description . . . . .	159
8.134.2 Field Documentation . . . . .	160
8.134.2.1 address . . . . .	160
8.134.2.2 port . . . . .	160
8.134.2.3 TlvPresent . . . . .	160
8.135loc_IPv6Config Struct Reference . . . . .	160
8.135.1 Detailed Description . . . . .	160
8.135.2 Field Documentation . . . . .	160
8.135.2.1 IPv6Addr . . . . .	160
8.135.2.2 IPv6Port . . . . .	160
8.136loc_IPv6Info Struct Reference . . . . .	160
8.136.1 Detailed Description . . . . .	161
8.136.2 Field Documentation . . . . .	161
8.136.2.1 address . . . . .	161
8.136.2.2 port . . . . .	161
8.136.2.3 TlvPresent . . . . .	161
8.137loc_LocApplicationInfo Struct Reference . . . . .	161

8.137.1 Detailed Description	161
8.137.2 Field Documentation	162
8.137.2.1 appNameLength	162
8.137.2.2 appProviderLength	162
8.137.2.3 appVersionLength	162
8.137.2.4 appVersionValid	162
8.137.2.5 pAppName	162
8.137.2.6 pAppProvider	162
8.137.2.7 pAppVersion	162
8.138loc_MinIntervalTlv Struct Reference	162
8.138.1 Detailed Description	162
8.138.2 Field Documentation	163
8.138.2.1 interval	163
8.138.2.2 TlvPresent	163
8.139loc_precisionDilution Struct Reference	163
8.139.1 Detailed Description	163
8.139.2 Field Documentation	163
8.139.2.1 HDOP	163
8.139.2.2 PDOP	163
8.139.2.3 VDOP	163
8.140loc_satelliteInfo Struct Reference	163
8.140.1 Detailed Description	164
8.140.2 Field Documentation	165
8.140.2.1 azimuth	165
8.140.2.2 elevation	165
8.140.2.3 gnssSvId	165
8.140.2.4 healthStatus	165
8.140.2.5 snr	165
8.140.2.6 svInfoMask	166
8.140.2.7 svListLen	166
8.140.2.8 svStatus	166
8.140.2.9 system	166
8.140.2.10validMask	166
8.141loc_sensorDataUsage Struct Reference	166
8.141.1 Detailed Description	166
8.141.2 Field Documentation	166
8.141.2.1 aidingIndicatorMask	166
8.141.2.2 usageMask	166
8.142loc_SV Struct Reference	166
8.142.1 Detailed Description	167

8.142.2 Field Documentation	167
8.142.2.1 id	167
8.142.2.2 mask	167
8.142.2.3 system	167
8.143loc_SVInfo Struct Reference	167
8.143.1 Detailed Description	167
8.143.2 Field Documentation	168
8.143.2.1 len	168
8.143.2.2 pSV	168
8.144loc_svUsedforFix Struct Reference	168
8.144.1 Detailed Description	168
8.144.2 Field Documentation	168
8.144.2.1 gnssSvUsedList	168
8.144.2.2 gnssSvUsedList_len	168
8.145loc_urlAddr Struct Reference	168
8.145.1 Detailed Description	169
8.145.2 Field Documentation	169
8.145.2.1 address	169
8.145.2.2 TlvPresent	169
8.146loc_URLAddrInfo Struct Reference	169
8.146.1 Detailed Description	169
8.146.2 Field Documentation	169
8.146.2.1 urlAddr	169
8.147lteSSInfo Struct Reference	169
8.147.1 Detailed Description	170
8.147.2 Field Documentation	170
8.147.2.1 rsrp	170
8.147.2.2 rsrq	170
8.147.2.3 rssi	170
8.147.2.4 snr	170
8.148messageModeTlv Struct Reference	170
8.148.1 Detailed Description	170
8.148.2 Field Documentation	171
8.148.2.1 MessageModelInfo	171
8.148.2.2 TlvPresent	171
8.149nas_acqOrderPref Struct Reference	171
8.149.1 Detailed Description	171
8.149.2 Field Documentation	171
8.149.2.1 acqOrdeLen	171
8.149.2.2 pAcqOrder	171

8.150nas_AcqOrderPrefTlv Struct Reference . . . . .	171
8.150.1 Detailed Description . . . . .	172
8.150.2 Field Documentation . . . . .	172
8.150.2.1 acqOrdeLen . . . . .	172
8.150.2.2 pAcqOrder . . . . .	172
8.150.2.3 TlvPresent . . . . .	172
8.151nas_ActPilotPNElement Struct Reference . . . . .	172
8.151.1 Detailed Description . . . . .	172
8.151.2 Field Documentation . . . . .	173
8.151.2.1 ActSetPilotPN . . . . .	173
8.151.2.2 ActSetPilotPNStrength . . . . .	173
8.152nas_AddCDMASysInfo Struct Reference . . . . .	173
8.152.1 Detailed Description . . . . .	173
8.152.2 Field Documentation . . . . .	173
8.152.2.1 geoSysIdx . . . . .	173
8.152.2.2 regPrd . . . . .	173
8.153nas_AddSysInfo Struct Reference . . . . .	173
8.153.1 Detailed Description . . . . .	173
8.153.2 Field Documentation . . . . .	174
8.153.2.1 cellBroadcastCap . . . . .	174
8.153.2.2 geoSysIdx . . . . .	174
8.154nas_BandPrefInfoTlv Struct Reference . . . . .	174
8.154.1 Detailed Description . . . . .	174
8.154.2 Field Documentation . . . . .	174
8.154.2.1 bits_129_192 . . . . .	174
8.154.2.2 bits_193_256 . . . . .	174
8.154.2.3 bits_1_64 . . . . .	175
8.154.2.4 bits_65_128 . . . . .	175
8.154.2.5 TlvPresent . . . . .	175
8.155nas_BandPrefTlv Struct Reference . . . . .	175
8.155.1 Detailed Description . . . . .	175
8.155.2 Field Documentation . . . . .	177
8.155.2.1 BandPref . . . . .	177
8.155.2.2 TlvPresent . . . . .	177
8.156nas_CallBarringSysInfo Struct Reference . . . . .	177
8.156.1 Detailed Description . . . . .	177
8.156.2 Field Documentation . . . . .	177
8.156.2.1 csBarStatus . . . . .	177
8.156.2.2 psBarStatus . . . . .	177
8.157nas_callBarStatus Struct Reference . . . . .	177

8.157.1 Detailed Description . . . . .	178
8.157.2 Field Documentation . . . . .	178
8.157.2.1 csBarStatus . . . . .	178
8.157.2.2 psBarStatus . . . . .	178
8.158nas_CDMAChannel Struct Reference . . . . .	178
8.158.1 Detailed Description . . . . .	178
8.158.2 Field Documentation . . . . .	179
8.158.2.1 priChA . . . . .	179
8.158.2.2 priChB . . . . .	179
8.158.2.3 secChA . . . . .	179
8.158.2.4 secChB . . . . .	179
8.159nas_CDMAECIOThresh Struct Reference . . . . .	179
8.159.1 Detailed Description . . . . .	179
8.159.2 Field Documentation . . . . .	179
8.159.2.1 CDMAECIOThreshListLen . . . . .	179
8.159.2.2 pCDMAECIOThreshList . . . . .	180
8.160nas_CDMAInfo Struct Reference . . . . .	180
8.160.1 Detailed Description . . . . .	180
8.160.2 Field Documentation . . . . .	180
8.160.2.1 baseId . . . . .	180
8.160.2.2 baseLat . . . . .	180
8.160.2.3 baseLong . . . . .	180
8.160.2.4 nid . . . . .	181
8.160.2.5 refpn . . . . .	181
8.160.2.6 sid . . . . .	181
8.161nas_CDMARSSIThresh Struct Reference . . . . .	181
8.161.1 Detailed Description . . . . .	181
8.161.2 Field Documentation . . . . .	181
8.161.2.1 CDMARSSIThreshListLen . . . . .	181
8.161.2.2 pCDMARSSIThreshList . . . . .	181
8.162nas_CDMA SysInfo Struct Reference . . . . .	181
8.162.1 Detailed Description . . . . .	182
8.162.2 Field Documentation . . . . .	184
8.162.2.1 baseId . . . . .	184
8.162.2.2 baseLat . . . . .	184
8.162.2.3 baseLong . . . . .	184
8.162.2.4 bsInfoValid . . . . .	184
8.162.2.5 bsPRev . . . . .	184
8.162.2.6 bsPRevValid . . . . .	184
8.162.2.7 ccsSupported . . . . .	184



8.162.2.8 ccsSupportedValid . . . . .	184
8.162.2.9 cdmaSysIdValid . . . . .	184
8.162.2.10sSysPriMatch . . . . .	184
8.162.2.11sSysPriMatchValid . . . . .	184
8.162.2.12MCC . . . . .	184
8.162.2.13MNC . . . . .	184
8.162.2.14networkID . . . . .	184
8.162.2.15networkIdValid . . . . .	185
8.162.2.16packetZone . . . . .	185
8.162.2.17packetZoneValid . . . . .	185
8.162.2.18pRevInUse . . . . .	185
8.162.2.19pRevInUseValid . . . . .	185
8.162.2.20sysInfoCDMA . . . . .	185
8.162.2.21systemID . . . . .	185
8.163nas_CDMASysInfoExt Struct Reference . . . . .	185
8.163.1 Detailed Description . . . . .	185
8.163.2 Field Documentation . . . . .	185
8.163.2.1 imsi_11_12 . . . . .	185
8.163.2.2 MCC . . . . .	185
8.164nas_cellParams Struct Reference . . . . .	185
8.164.1 Detailed Description . . . . .	186
8.164.2 Field Documentation . . . . .	186
8.164.2.1 pci . . . . .	186
8.164.2.2 rsrp . . . . .	186
8.164.2.3 rsrq . . . . .	186
8.164.2.4 rssi . . . . .	186
8.164.2.5 srxlev . . . . .	186
8.165nas_ciotAcqOrderPref Struct Reference . . . . .	186
8.165.1 Detailed Description . . . . .	186
8.165.2 Field Documentation . . . . .	187
8.165.2.1 ciotAcqOrderLen . . . . .	187
8.165.2.2 pCiotAcqOrder . . . . .	187
8.166nas_CiotAcqOrderPrefTlv Struct Reference . . . . .	187
8.166.1 Detailed Description . . . . .	187
8.166.2 Field Documentation . . . . .	188
8.166.2.1 ciotAcqOrderLen . . . . .	188
8.166.2.2 pCiotAcqOrder . . . . .	188
8.166.2.3 TlvPresent . . . . .	188
8.167nas_CiotLteOpModePrefTlv Struct Reference . . . . .	188
8.167.1 Detailed Description . . . . .	188

8.167.2 Field Documentation . . . . .	188
8.167.2.1 ciotLteOpModePref . . . . .	188
8.167.2.2 TlvPresent . . . . .	188
8.168nas_CommInfo Struct Reference . . . . .	188
8.168.1 Detailed Description . . . . .	189
8.168.2 Field Documentation . . . . .	190
8.168.2.1 imsRegState . . . . .	190
8.168.2.2 modemMode . . . . .	190
8.168.2.3 psState . . . . .	190
8.168.2.4 systemMode . . . . .	190
8.168.2.5 temperature . . . . .	190
8.169nas_CsgId Struct Reference . . . . .	190
8.169.1 Detailed Description . . . . .	190
8.169.2 Field Documentation . . . . .	190
8.169.2.1 csgId . . . . .	190
8.169.2.2 TlvPresent . . . . .	190
8.170nas_CSGID Struct Reference . . . . .	190
8.170.1 Detailed Description . . . . .	191
8.170.2 Field Documentation . . . . .	191
8.170.2.1 id . . . . .	191
8.170.2.2 mcc . . . . .	191
8.170.2.3 mnc . . . . .	191
8.170.2.4 mncPcsDigits . . . . .	191
8.170.2.5 rat . . . . .	191
8.171nas_currentPLMN Struct Reference . . . . .	191
8.171.1 Detailed Description . . . . .	192
8.171.2 Field Documentation . . . . .	192
8.171.2.1 MCC . . . . .	192
8.171.2.2 MNC . . . . .	192
8.171.2.3 netDescr . . . . .	192
8.171.2.4 netDescrLength . . . . .	192
8.172nas_dataSrvCapabilities Struct Reference . . . . .	192
8.172.1 Detailed Description . . . . .	192
8.172.2 Field Documentation . . . . .	193
8.172.2.1 dataCapabilities . . . . .	193
8.172.2.2 dataCapabilitiesLen . . . . .	193
8.173nas_DataStatusDetail Struct Reference . . . . .	193
8.173.1 Detailed Description . . . . .	193
8.173.2 Field Documentation . . . . .	194
8.173.2.1 IPAddress . . . . .	194

8.173.2.2 LastErrCode . . . . .	194
8.174nas_detailSvcInfo Struct Reference . . . . .	195
8.174.1 Detailed Description . . . . .	195
8.174.2 Field Documentation . . . . .	196
8.174.2.1 hdrHybrid . . . . .	196
8.174.2.2 hdrSrvStatus . . . . .	196
8.174.2.3 isSysForbidden . . . . .	196
8.174.2.4 srvCapability . . . . .	196
8.174.2.5 srvStatus . . . . .	196
8.175nas_DeviceConfigDetail Struct Reference . . . . .	196
8.175.1 Detailed Description . . . . .	196
8.175.2 Field Documentation . . . . .	197
8.175.2.1 Chipset . . . . .	197
8.175.2.2 HWVersion . . . . .	197
8.175.2.3 QLIC . . . . .	197
8.175.2.4 Technology . . . . .	197
8.176nas_dirNum Struct Reference . . . . .	197
8.176.1 Detailed Description . . . . .	197
8.176.2 Field Documentation . . . . .	197
8.176.2.1 dirNum . . . . .	197
8.176.2.2 dirNumLen . . . . .	197
8.177nas_DRCPParams Struct Reference . . . . .	197
8.177.1 Detailed Description . . . . .	198
8.177.2 Field Documentation . . . . .	198
8.177.2.1 DRCCover . . . . .	198
8.177.2.2 DRCValue . . . . .	198
8.178nas_ecioListElement Struct Reference . . . . .	198
8.178.1 Detailed Description . . . . .	198
8.178.2 Field Documentation . . . . .	198
8.178.2.1 ecio . . . . .	199
8.178.2.2 radiolf . . . . .	199
8.179nas_ECIOThresh Struct Reference . . . . .	199
8.179.1 Detailed Description . . . . .	199
8.179.2 Field Documentation . . . . .	199
8.179.2.1 ECIOThresListLen . . . . .	199
8.179.2.2 pECIOThresList . . . . .	199
8.180nas_EdrxCiotLteMode Struct Reference . . . . .	199
8.180.1 Detailed Description . . . . .	199
8.180.2 Field Documentation . . . . .	200
8.180.2.1 lteOpMode . . . . .	200

8.180.2.2 TlvPresent . . . . .	200
8.181nas_EdrxCycleLength Struct Reference . . . . .	200
8.181.1 Detailed Description . . . . .	200
8.181.2 Field Documentation . . . . .	200
8.181.2.1 cycleLength . . . . .	200
8.181.2.2 TlvPresent . . . . .	200
8.182nas_EdrxEnableType Struct Reference . . . . .	200
8.182.1 Detailed Description . . . . .	201
8.182.2 Field Documentation . . . . .	201
8.182.2.1 edrxEnabled . . . . .	201
8.182.2.2 TlvPresent . . . . .	201
8.183nas_EdrxPagingTimeWindow Struct Reference . . . . .	201
8.183.1 Detailed Description . . . . .	201
8.183.2 Field Documentation . . . . .	201
8.183.2.1 edrxPtw . . . . .	201
8.183.2.2 TlvPresent . . . . .	201
8.184nas_EdrxRatType Struct Reference . . . . .	201
8.184.1 Detailed Description . . . . .	202
8.184.2 Field Documentation . . . . .	202
8.184.2.1 edrxRatType . . . . .	202
8.184.2.2 TlvPresent . . . . .	202
8.185nas_EmerModeTlv Struct Reference . . . . .	202
8.185.1 Detailed Description . . . . .	202
8.185.2 Field Documentation . . . . .	203
8.185.2.1 EmerMode . . . . .	203
8.185.2.2 TlvPresent . . . . .	203
8.186nas_errorRateListElement Struct Reference . . . . .	203
8.186.1 Detailed Description . . . . .	203
8.186.2 Field Documentation . . . . .	204
8.186.2.1 errorRate . . . . .	204
8.186.2.2 radiolf . . . . .	204
8.187nas_ForbiddenNetworks3GPP Struct Reference . . . . .	204
8.187.1 Detailed Description . . . . .	204
8.187.2 Field Documentation . . . . .	205
8.187.2.1 forbiddenNwInstLen . . . . .	205
8.187.2.2 MCC . . . . .	205
8.187.2.3 MNC . . . . .	205
8.187.2.4 TlvPresent . . . . .	205
8.188nas_GERANInfo Struct Reference . . . . .	205
8.188.1 Detailed Description . . . . .	205

8.188.2 Field Documentation	206
8.188.2.1 arfcn	206
8.188.2.2 bsic	206
8.188.2.3 cellID	206
8.188.2.4 insNmrCellInfo	206
8.188.2.5 lac	206
8.188.2.6 nmrlnst	206
8.188.2.7 plmn	206
8.188.2.8 rxLev	206
8.188.2.9 timingAdvance	206
8.189nas_geranInstArr Struct Reference	206
8.189.1 Detailed Description	207
8.189.2 Field Documentation	207
8.189.2.1 geranArfcn	207
8.189.2.2 geranBsicBcc	207
8.189.2.3 geranBsicNcc	207
8.189.2.4 geranRank	207
8.189.2.5 geranRssi	207
8.190nas_geranInstInfo Struct Reference	207
8.190.1 Detailed Description	207
8.190.2 Field Documentation	208
8.190.2.1 geranArfcn	208
8.190.2.2 geranBsicBcc	208
8.190.2.3 geranBsicNcc	208
8.190.2.4 geranRssi	208
8.191nas_gsmCellInfo Struct Reference	208
8.191.1 Detailed Description	208
8.191.2 Field Documentation	209
8.191.2.1 arfcn	209
8.191.2.2 band1900	209
8.191.2.3 bsicld	209
8.191.2.4 cellIdValid	209
8.191.2.5 rssi	209
8.191.2.6 srxlev	209
8.192nas_GSMRSSIthresh Struct Reference	209
8.192.1 Detailed Description	209
8.192.2 Field Documentation	210
8.192.2.1 GSMRSSIthreshListLen	210
8.192.2.2 pGSMRSSIthreshList	210
8.193nas_GSMSrvStatusInfo Struct Reference	210

8.193.1 Detailed Description . . . . .	210
8.193.2 Field Documentation . . . . .	210
8.193.2.1 isPrefDataPath . . . . .	210
8.193.2.2 srvStatus . . . . .	210
8.193.2.3 trueSrvStatus . . . . .	211
8.194nas_GSMSysInfo Struct Reference . . . . .	211
8.194.1 Detailed Description . . . . .	211
8.194.2 Field Documentation . . . . .	213
8.194.2.1 cellId . . . . .	213
8.194.2.2 cellIdValid . . . . .	213
8.194.2.3 dtmSupp . . . . .	213
8.194.2.4 dtmSuppValid . . . . .	213
8.194.2.5 egprsSupp . . . . .	213
8.194.2.6 egprsSuppValid . . . . .	213
8.194.2.7 lac . . . . .	213
8.194.2.8 lacValid . . . . .	213
8.194.2.9 MCC . . . . .	213
8.194.2.10MNC . . . . .	213
8.194.2.11networkIdValid . . . . .	213
8.194.2.12regRejectInfoValid . . . . .	213
8.194.2.13rejCause . . . . .	213
8.194.2.14rejectSrvDomain . . . . .	213
8.194.2.15sysInfoGSM . . . . .	213
8.195nas_GWAcqOrderPrefTlv Struct Reference . . . . .	213
8.195.1 Detailed Description . . . . .	214
8.195.2 Field Documentation . . . . .	214
8.195.2.1 GWAcqOrderPref . . . . .	214
8.195.2.2 TlvPresent . . . . .	214
8.196nas_HDRECIOThresh Struct Reference . . . . .	214
8.196.1 Detailed Description . . . . .	214
8.196.2 Field Documentation . . . . .	214
8.196.2.1 HDRECIOThreshListLen . . . . .	214
8.196.2.2 pHDRECIOThreshList . . . . .	214
8.197nas_HDRIOThresh Struct Reference . . . . .	214
8.197.1 Detailed Description . . . . .	215
8.197.2 Field Documentation . . . . .	215
8.197.2.1 HDRIOThreshListLen . . . . .	215
8.197.2.2 pHDRIOThreshList . . . . .	215
8.198nas_HDRPersonality_Ind_Data Struct Reference . . . . .	215
8.198.1 Detailed Description . . . . .	215

8.198.2 Field Documentation . . . . .	216
8.198.2.1 pCurrentPersonality . . . . .	216
8.198.2.2 pPersonalityListLength . . . . .	216
8.198.2.3 pProtocolSubtypeElement . . . . .	216
8.199nas_HDRRSSIThresh Struct Reference . . . . .	216
8.199.1 Detailed Description . . . . .	216
8.199.2 Field Documentation . . . . .	216
8.199.2.1 HDRRSSIThreshListLen . . . . .	216
8.199.2.2 pHDRRSSIThreshList . . . . .	216
8.200nas_HDRSINRThresh Struct Reference . . . . .	216
8.200.1 Detailed Description . . . . .	216
8.200.2 Field Documentation . . . . .	217
8.200.2.1 HDRSINRThresListLen . . . . .	217
8.200.2.2 pHDRSINRThresList . . . . .	217
8.201nas_HDRSINRThreshold Struct Reference . . . . .	217
8.201.1 Detailed Description . . . . .	217
8.201.2 Field Documentation . . . . .	217
8.201.2.1 HDRSINRThreshListLen . . . . .	217
8.201.2.2 pHDRSINRThreshList . . . . .	218
8.202nas_HDRSysInfo Struct Reference . . . . .	218
8.202.1 Detailed Description . . . . .	218
8.202.2 Field Documentation . . . . .	219
8.202.2.1 hdrActiveProt . . . . .	219
8.202.2.2 hdrActiveProtValid . . . . .	219
8.202.2.3 hdrPersonality . . . . .	219
8.202.2.4 hdrPersonalityValid . . . . .	219
8.202.2.5 is856SysId . . . . .	219
8.202.2.6 is856SysIdValid . . . . .	219
8.202.2.7 isSysPrIMatch . . . . .	219
8.202.2.8 isSysPrIMatchValid . . . . .	219
8.202.2.9 sysInfoHDR . . . . .	220
8.203nas_homeNwMNC3GppTlv Struct Reference . . . . .	220
8.203.1 Detailed Description . . . . .	220
8.203.2 Field Documentation . . . . .	220
8.203.2.1 is3GppNw . . . . .	220
8.203.2.2 mccIncPcsDigit . . . . .	220
8.203.2.3 TlvPresent . . . . .	220
8.204nas_homeSIDNID Struct Reference . . . . .	220
8.204.1 Detailed Description . . . . .	221
8.204.2 Field Documentation . . . . .	221

8.204.2.1 numInstances . . . . .	221
8.204.2.2 SidNid . . . . .	221
8.205nas_ImVoiceSupportLteTlv Struct Reference . . . . .	221
8.205.1 Detailed Description . . . . .	221
8.205.2 Field Documentation . . . . .	221
8.205.2.1 ImVoiceSupportLte . . . . .	221
8.205.2.2 TLVPresent . . . . .	221
8.206nas_infoInterFreq Struct Reference . . . . .	221
8.206.1 Detailed Description . . . . .	222
8.206.2 Field Documentation . . . . .	222
8.206.2.1 cell_resel_priority . . . . .	222
8.206.2.2 cellInterFreqParams . . . . .	222
8.206.2.3 cells_len . . . . .	222
8.206.2.4 earfcn . . . . .	222
8.206.2.5 threshXHigh . . . . .	223
8.206.2.6 threshXLow . . . . .	223
8.207nas_IOTresh Struct Reference . . . . .	223
8.207.1 Detailed Description . . . . .	223
8.207.2 Field Documentation . . . . .	223
8.207.2.1 IOThesListLen . . . . .	223
8.207.2.2 pIOThesList . . . . .	223
8.208nas_lteBandPrefExt Struct Reference . . . . .	223
8.208.1 Detailed Description . . . . .	223
8.208.2 Field Documentation . . . . .	224
8.208.2.1 bits_129_192 . . . . .	224
8.208.2.2 bits_193_256 . . . . .	224
8.208.2.3 bits_1_64 . . . . .	224
8.208.2.4 bits_65_128 . . . . .	224
8.209nas_LTEBandPrefTlv Struct Reference . . . . .	224
8.209.1 Detailed Description . . . . .	224
8.209.2 Field Documentation . . . . .	225
8.209.2.1 LTEBandPref . . . . .	226
8.209.2.2 TlvPresent . . . . .	226
8.210nas_LteCiotOpModeTlv Struct Reference . . . . .	226
8.210.1 Detailed Description . . . . .	226
8.210.2 Field Documentation . . . . .	226
8.210.2.1 campedCiotLteOpMode . . . . .	226
8.210.2.2 TlvPresent . . . . .	226
8.211nas_lteEARFCN Struct Reference . . . . .	226
8.211.1 Detailed Description . . . . .	226



8.211.2 Field Documentation . . . . .	227
8.211.2.1 earfcn0 . . . . .	227
8.211.2.2 earfcn1 . . . . .	227
8.211.2.3 status . . . . .	227
8.212nas_LteEarfcnInfo Struct Reference . . . . .	227
8.212.1 Detailed Description . . . . .	227
8.212.2 Field Documentation . . . . .	227
8.212.2.1 lteEarfcn . . . . .	227
8.212.2.2 lteInterEarfcnlen . . . . .	227
8.212.2.3 TlvPresent . . . . .	227
8.213nas_LteEmbmsCoverageTlv Struct Reference . . . . .	227
8.213.1 Detailed Description . . . . .	228
8.213.2 Field Documentation . . . . .	228
8.213.2.1 LteEmbmsCoverage . . . . .	228
8.213.2.2 TLVPresent . . . . .	228
8.214nas_LteEmbmsTraceIdTlv Struct Reference . . . . .	228
8.214.1 Detailed Description . . . . .	228
8.214.2 Field Documentation . . . . .	228
8.214.2.1 LteEmbmsTraceId . . . . .	228
8.214.2.2 TLVPresent . . . . .	229
8.215nas_LteGsmCellInfo Struct Reference . . . . .	229
8.215.1 Detailed Description . . . . .	229
8.215.2 Field Documentation . . . . .	229
8.215.2.1 cellReselPriority . . . . .	229
8.215.2.2 cells_len . . . . .	230
8.215.2.3 GsmCellInfo . . . . .	230
8.215.2.4 nccPermitted . . . . .	230
8.215.2.5 threshGsmHigh . . . . .	230
8.215.2.6 threshGsmLow . . . . .	230
8.216nas_LTEInfo Struct Reference . . . . .	230
8.216.1 Detailed Description . . . . .	230
8.216.2 Field Documentation . . . . .	231
8.216.2.1 band . . . . .	231
8.216.2.2 bandwidth . . . . .	232
8.216.2.3 emmConnState . . . . .	232
8.216.2.4 emmState . . . . .	232
8.216.2.5 emmSubState . . . . .	232
8.216.2.6 RXChan . . . . .	232
8.216.2.7 TXChan . . . . .	232
8.217nas_LTEInfoInterfreq Struct Reference . . . . .	232

8.217.1 Detailed Description . . . . .	232
8.217.2 Field Documentation . . . . .	232
8.217.2.1 freqsLen . . . . .	232
8.217.2.2 InfoInterfreq . . . . .	232
8.217.2.3 ueInIdle . . . . .	232
8.218nas_LTEInfoIntrafreq Struct Reference . . . . .	232
8.218.1 Detailed Description . . . . .	233
8.218.2 Field Documentation . . . . .	234
8.218.2.1 CellParams . . . . .	234
8.218.2.2 cellReselPriority . . . . .	234
8.218.2.3 cellsLen . . . . .	234
8.218.2.4 earfcn . . . . .	234
8.218.2.5 globalCellId . . . . .	234
8.218.2.6 plmn . . . . .	234
8.218.2.7 servingCellId . . . . .	234
8.218.2.8 sIntraSearch . . . . .	234
8.218.2.9 sNonIntraSearch . . . . .	234
8.218.2.10tac . . . . .	234
8.218.2.11threshServingLow . . . . .	234
8.218.2.12ueInIdle . . . . .	234
8.219nas_LTEInfoNeighboringGSM Struct Reference . . . . .	234
8.219.1 Detailed Description . . . . .	235
8.219.2 Field Documentation . . . . .	235
8.219.2.1 freqsLen . . . . .	235
8.219.2.2 LteGsmCellInfo . . . . .	235
8.219.2.3 ueInIdle . . . . .	235
8.220nas_LTEInfoNeighboringWCDMA Struct Reference . . . . .	235
8.220.1 Detailed Description . . . . .	235
8.220.2 Field Documentation . . . . .	236
8.220.2.1 freqsLen . . . . .	236
8.220.2.2 LTEWCDMACellInfo . . . . .	236
8.220.2.3 ueInIdle . . . . .	236
8.221nas_LteM1BandPrefTlv Struct Reference . . . . .	236
8.221.1 Detailed Description . . . . .	236
8.221.2 Field Documentation . . . . .	237
8.221.2.1 lteM1BandPref . . . . .	238
8.221.2.2 TlvPresent . . . . .	238
8.222nas_LteNb1BandPrefTlv Struct Reference . . . . .	238
8.222.1 Detailed Description . . . . .	238
8.222.2 Field Documentation . . . . .	239

8.222.2.1 lteNb1BandPref . . . . .	240
8.222.2.2 TlvPresent . . . . .	240
8.223nas_LTEOperationalModeTlv Struct Reference . . . . .	240
8.223.1 Detailed Description . . . . .	240
8.223.2 Field Documentation . . . . .	240
8.223.2.1 ciotLteOpMode . . . . .	240
8.223.2.2 TlvPresent . . . . .	240
8.224nas_LTEOperationMode Struct Reference . . . . .	240
8.224.1 Detailed Description . . . . .	240
8.224.2 Field Documentation . . . . .	241
8.224.2.1 pLTEOperationMode . . . . .	241
8.224.2.2 TlvPresent . . . . .	241
8.225nas_LteOpMode Struct Reference . . . . .	241
8.225.1 Detailed Description . . . . .	241
8.225.2 Field Documentation . . . . .	241
8.225.2.1 lteOpMode . . . . .	241
8.225.2.2 TlvPresent . . . . .	241
8.226nas_lteOpModeTlv Struct Reference . . . . .	241
8.226.1 Detailed Description . . . . .	242
8.226.2 Field Documentation . . . . .	242
8.226.2.1 lteOpMode . . . . .	242
8.226.2.2 lteOpModeLen . . . . .	242
8.226.2.3 MCC . . . . .	242
8.226.2.4 MNC . . . . .	242
8.226.2.5 TlvPresent . . . . .	242
8.227nas_ltePCI Struct Reference . . . . .	242
8.227.1 Detailed Description . . . . .	243
8.227.2 Field Documentation . . . . .	243
8.227.2.1 earfcn . . . . .	243
8.227.2.2 pci . . . . .	243
8.227.2.3 status . . . . .	243
8.228nas_LteRegDomainTlv Struct Reference . . . . .	243
8.228.1 Detailed Description . . . . .	243
8.228.2 Field Documentation . . . . .	244
8.228.2.1 LteRegDomain . . . . .	244
8.228.2.2 TLVPresent . . . . .	244
8.229nas_lteRsrpInformation Struct Reference . . . . .	244
8.229.1 Detailed Description . . . . .	244
8.229.2 Field Documentation . . . . .	244
8.229.2.1 rsrplevel . . . . .	244

8.230nas_LTERSRPThresh Struct Reference . . . . .	244
8.230.1 Detailed Description . . . . .	244
8.230.2 Field Documentation . . . . .	245
8.230.2.1 LTERSRPThreshListLen . . . . .	245
8.230.2.2 pLTERSRPThreshList . . . . .	245
8.231nas_LTERSRQThresh Struct Reference . . . . .	245
8.231.1 Detailed Description . . . . .	245
8.231.2 Field Documentation . . . . .	245
8.231.2.1 LTERSRQThreshListLen . . . . .	245
8.231.2.2 pLTERSRQThreshList . . . . .	245
8.232nas_LTERSSIThresh Struct Reference . . . . .	245
8.232.1 Detailed Description . . . . .	245
8.232.2 Field Documentation . . . . .	246
8.232.2.1 LTERSSIThreshListLen . . . . .	246
8.232.2.2 pLTERSSIThreshList . . . . .	246
8.233nas_LTESigRptCfg Struct Reference . . . . .	246
8.233.1 Detailed Description . . . . .	246
8.233.2 Field Documentation . . . . .	247
8.233.2.1 avgPeriod . . . . .	247
8.233.2.2 rptRate . . . . .	247
8.234nas_LTESigRptConfig Struct Reference . . . . .	247
8.234.1 Detailed Description . . . . .	247
8.234.2 Field Documentation . . . . .	248
8.234.2.1 avgPeriod . . . . .	248
8.234.2.2 rptRate . . . . .	248
8.235nas_lteSnrinformation Struct Reference . . . . .	248
8.235.1 Detailed Description . . . . .	248
8.235.2 Field Documentation . . . . .	248
8.235.2.1 snrlevel . . . . .	248
8.236nas_LTESNRThresh Struct Reference . . . . .	248
8.236.1 Detailed Description . . . . .	249
8.236.2 Field Documentation . . . . .	249
8.236.2.1 LTESNRThresListLen . . . . .	249
8.236.2.2 pLTESNRThresList . . . . .	249
8.237nas_LTESNRThreshold Struct Reference . . . . .	249
8.237.1 Detailed Description . . . . .	249
8.237.2 Field Documentation . . . . .	249
8.237.2.1 LTESNRThreshListLen . . . . .	249
8.237.2.2 pLTESNRThreshList . . . . .	250
8.238nas_LTESysInfo Struct Reference . . . . .	250

8.238.1 Detailed Description . . . . .	250
8.238.2 Field Documentation . . . . .	251
8.238.2.1 cellId . . . . .	252
8.238.2.2 cellIdValid . . . . .	252
8.238.2.3 lac . . . . .	252
8.238.2.4 lacValid . . . . .	252
8.238.2.5 MCC . . . . .	252
8.238.2.6 MNC . . . . .	252
8.238.2.7 networkIdValid . . . . .	252
8.238.2.8 regRejectInfoValid . . . . .	252
8.238.2.9 rejCause . . . . .	252
8.238.2.10 rejectSrvDomain . . . . .	252
8.238.2.11 sysInfoLTE . . . . .	252
8.238.2.12 tac . . . . .	252
8.238.2.13 tacValid . . . . .	252
8.239 nas_LteVoiceDomainTlv Struct Reference . . . . .	252
8.239.1 Detailed Description . . . . .	252
8.239.2 Field Documentation . . . . .	252
8.239.2.1 LteVoiceDomain . . . . .	253
8.239.2.2 TLVPresent . . . . .	253
8.240 nas_LteWcdmaCellInfo Struct Reference . . . . .	253
8.240.1 Detailed Description . . . . .	253
8.240.2 Field Documentation . . . . .	253
8.240.2.1 cellReselPriority . . . . .	253
8.240.2.2 cellsLen . . . . .	253
8.240.2.3 threshXhigh . . . . .	254
8.240.2.4 threshXlow . . . . .	254
8.240.2.5 uarfcn . . . . .	254
8.240.2.6 WCDMACellInfo . . . . .	254
8.241 nas_Mdn Struct Reference . . . . .	254
8.241.1 Detailed Description . . . . .	254
8.241.2 Field Documentation . . . . .	254
8.241.2.1 mdn . . . . .	254
8.241.2.2 mdnLen . . . . .	254
8.242 nas_minBasedIMSI Struct Reference . . . . .	254
8.242.1 Detailed Description . . . . .	254
8.242.2 Field Documentation . . . . .	255
8.242.2.1 imsiM1112 . . . . .	255
8.242.2.2 imsiMS1 . . . . .	255
8.242.2.3 imsiMS2 . . . . .	255

8.242.2.4 mccM . . . . .	255
8.243nas_MNCPGSDigitStatus Struct Reference . . . . .	255
8.243.1 Detailed Description . . . . .	255
8.243.2 Field Documentation . . . . .	256
8.243.2.1 MCC . . . . .	256
8.243.2.2 MNC . . . . .	256
8.243.2.3 MNCIncPGSDigit . . . . .	256
8.243.2.4 TLVPresent . . . . .	256
8.244nas_MNRInfo Struct Reference . . . . .	256
8.244.1 Detailed Description . . . . .	256
8.244.2 Field Documentation . . . . .	256
8.244.2.1 mcc . . . . .	256
8.244.2.2 mnc . . . . .	256
8.244.2.3 rat . . . . .	256
8.245nas_ModePrefTlv Struct Reference . . . . .	256
8.245.1 Detailed Description . . . . .	257
8.245.2 Field Documentation . . . . .	257
8.245.2.1 ModePref . . . . .	257
8.245.2.2 TlvPresent . . . . .	257
8.246nas_namName Struct Reference . . . . .	257
8.246.1 Detailed Description . . . . .	257
8.246.2 Field Documentation . . . . .	258
8.246.2.1 namName . . . . .	258
8.246.2.2 namNameLen . . . . .	258
8.247nas_netSelectionPref Struct Reference . . . . .	258
8.247.1 Detailed Description . . . . .	258
8.247.2 Field Documentation . . . . .	258
8.247.2.1 mcc . . . . .	258
8.247.2.2 mnc . . . . .	258
8.247.2.3 netReg . . . . .	258
8.248nas_NetSelPrefTlv Struct Reference . . . . .	258
8.248.1 Detailed Description . . . . .	258
8.248.2 Field Documentation . . . . .	259
8.248.2.1 NetSelPref . . . . .	259
8.248.2.2 TlvPresent . . . . .	259
8.249nas_networkNameSrcTlv Struct Reference . . . . .	259
8.249.1 Detailed Description . . . . .	259
8.249.2 Field Documentation . . . . .	259
8.249.2.1 nwNameSrc . . . . .	260
8.249.2.2 nwNameSrcLen . . . . .	260

8.249.2.3 TlvPresent . . . . .	260
8.250nas_NetworkStat1x Struct Reference . . . . .	260
8.250.1 Detailed Description . . . . .	260
8.250.2 Field Documentation . . . . .	261
8.250.2.1 ActSetCnt . . . . .	261
8.250.2.2 NeighborSetCnt . . . . .	261
8.250.2.3 pActPilotPNElements . . . . .	261
8.250.2.4 pNeighborSetPilotPN . . . . .	262
8.250.2.5 RX_EC_IO . . . . .	262
8.250.2.6 RX_PWR . . . . .	262
8.250.2.7 SO . . . . .	262
8.250.2.8 State . . . . .	262
8.250.2.9 TX_PWR . . . . .	262
8.251nas_NetworkStatEVDO Struct Reference . . . . .	262
8.251.1 Detailed Description . . . . .	262
8.251.2 Field Documentation . . . . .	263
8.251.2.1 MACIndex . . . . .	263
8.251.2.2 PER . . . . .	263
8.251.2.3 PilotEnergy . . . . .	263
8.251.2.4 pSectorID . . . . .	263
8.251.2.5 RX_PWR . . . . .	263
8.251.2.6 SectorIDLen . . . . .	263
8.251.2.7 SNR . . . . .	263
8.251.2.8 State . . . . .	263
8.252nas_nmrCellInfo Struct Reference . . . . .	263
8.252.1 Detailed Description . . . . .	264
8.252.2 Field Documentation . . . . .	264
8.252.2.1 nmrArfcn . . . . .	264
8.252.2.2 nmrBsic . . . . .	264
8.252.2.3 nmrCellID . . . . .	264
8.252.2.4 nmrLac . . . . .	264
8.252.2.5 nmrPlmn . . . . .	265
8.252.2.6 nmrRxLev . . . . .	265
8.253nas_nr5gBandPref Struct Reference . . . . .	265
8.253.1 Detailed Description . . . . .	265
8.253.2 Field Documentation . . . . .	265
8.253.2.1 bits_129_192 . . . . .	265
8.253.2.2 bits_193_256 . . . . .	265
8.253.2.3 bits_1_64 . . . . .	265
8.253.2.4 bits_65_128 . . . . .	265

8.254nas_NR5GCellStatusInfoTlv Struct Reference . . . . .	265
8.254.1 Detailed Description . . . . .	265
8.254.2 Field Documentation . . . . .	266
8.254.2.1 nr5gCellStatus . . . . .	266
8.254.2.2 TLVPresent . . . . .	266
8.255nas_NR5GCellStatusTlv Struct Reference . . . . .	266
8.255.1 Detailed Description . . . . .	266
8.255.2 Field Documentation . . . . .	266
8.255.2.1 nr5gCellStatus . . . . .	266
8.255.2.2 TlvPresent . . . . .	267
8.256nas_NR5GSerStatTlv Struct Reference . . . . .	267
8.256.1 Detailed Description . . . . .	267
8.256.2 Field Documentation . . . . .	267
8.256.2.1 isPrefDataPath . . . . .	267
8.256.2.2 srvStatus . . . . .	267
8.256.2.3 TlvPresent . . . . .	268
8.256.2.4 trueSrvStatus . . . . .	268
8.257nas_NR5GSrvStatusTlv Struct Reference . . . . .	268
8.257.1 Detailed Description . . . . .	268
8.257.2 Field Documentation . . . . .	268
8.257.2.1 isPrefDataPath . . . . .	268
8.257.2.2 srvStatus . . . . .	269
8.257.2.3 TLVPresent . . . . .	269
8.257.2.4 trueSrvStatus . . . . .	269
8.258nas_NR5GSysInfoTlv Struct Reference . . . . .	269
8.258.1 Detailed Description . . . . .	269
8.258.2 Field Documentation . . . . .	271
8.258.2.1 cellId . . . . .	271
8.258.2.2 cellIdValid . . . . .	271
8.258.2.3 isSysForbidden . . . . .	271
8.258.2.4 isSysForbiddenValid . . . . .	271
8.258.2.5 lac . . . . .	271
8.258.2.6 lacValid . . . . .	272
8.258.2.7 mcc . . . . .	272
8.258.2.8 mnc . . . . .	272
8.258.2.9 nwIdValid . . . . .	272
8.258.2.10regRejectInfoValid . . . . .	272
8.258.2.11rejectCause . . . . .	272
8.258.2.12rejectSrvDomain . . . . .	272
8.258.2.13roamStatus . . . . .	272



8.258.2.14roamStatusValid . . . . .	272
8.258.2.15srvCapability . . . . .	272
8.258.2.16srvCapabilityValid . . . . .	272
8.258.2.17srvDomain . . . . .	272
8.258.2.18srvDomainValid . . . . .	272
8.258.2.19tac . . . . .	272
8.258.2.20tacValid . . . . .	272
8.258.2.21TLVPresent . . . . .	272
8.259nas_NR5GSystemInfoTlv Struct Reference . . . . .	272
8.259.1 Detailed Description . . . . .	273
8.259.2 Field Documentation . . . . .	275
8.259.2.1 cellId . . . . .	275
8.259.2.2 cellIdValid . . . . .	275
8.259.2.3 lac . . . . .	275
8.259.2.4 lacValid . . . . .	275
8.259.2.5 MCC . . . . .	275
8.259.2.6 MNC . . . . .	275
8.259.2.7 nwIdValid . . . . .	275
8.259.2.8 regRejectInfoValid . . . . .	275
8.259.2.9 rejCause . . . . .	275
8.259.2.10rejectSrvDomain . . . . .	275
8.259.2.11roamStatus . . . . .	275
8.259.2.12roamStatusValid . . . . .	276
8.259.2.13srvcapability . . . . .	276
8.259.2.14srvCapValid . . . . .	276
8.259.2.15srvDomain . . . . .	276
8.259.2.16srvDomainValid . . . . .	276
8.259.2.17sysForbidden . . . . .	276
8.259.2.18sysForbiddenValid . . . . .	276
8.259.2.19tac . . . . .	276
8.259.2.20tacValid . . . . .	276
8.259.2.21TlvPresent . . . . .	276
8.260nas_NumScellsConfig Struct Reference . . . . .	276
8.260.1 Detailed Description . . . . .	276
8.260.2 Field Documentation . . . . .	276
8.260.2.1 numScellsCfg . . . . .	276
8.260.2.2 TlvPresent . . . . .	276
8.261nas_nwNameSrc3GppTlv Struct Reference . . . . .	276
8.261.1 Detailed Description . . . . .	277
8.261.2 Field Documentation . . . . .	277

8.261.2.1 NwNameSrc3Gpp . . . . .	277
8.261.2.2 TlvPresent . . . . .	277
8.262nas_operatorNameString Struct Reference . . . . .	277
8.262.1 Detailed Description . . . . .	277
8.262.2 Field Documentation . . . . .	277
8.262.2.1 PLMNName . . . . .	277
8.263nas_OperatorPLMNData Struct Reference . . . . .	277
8.263.1 Detailed Description . . . . .	278
8.263.2 Field Documentation . . . . .	278
8.263.2.1 lac1 . . . . .	278
8.263.2.2 lac2 . . . . .	278
8.263.2.3 mcc . . . . .	278
8.263.2.4 mnc . . . . .	278
8.263.2.5 PLMNRecID . . . . .	278
8.264nas_operatorPLMNList Struct Reference . . . . .	278
8.264.1 Detailed Description . . . . .	278
8.264.2 Field Documentation . . . . .	279
8.264.2.1 numInstance . . . . .	279
8.264.2.2 PLMNData . . . . .	279
8.265nas_PhyCaAggDIBW Struct Reference . . . . .	279
8.265.1 Detailed Description . . . . .	279
8.265.2 Field Documentation . . . . .	279
8.265.2.1 aggDIBW . . . . .	279
8.265.2.2 TlvPresent . . . . .	279
8.266nas_PhyCaAggPcellInfo Struct Reference . . . . .	279
8.266.1 Detailed Description . . . . .	280
8.266.2 Field Documentation . . . . .	280
8.266.2.1 dl_bw_value . . . . .	280
8.266.2.2 freq . . . . .	280
8.266.2.3 iLTEbandValue . . . . .	280
8.266.2.4 pci . . . . .	280
8.266.2.5 TlvPresent . . . . .	280
8.267nas_PhyCaAggScellDIBw Struct Reference . . . . .	280
8.267.1 Detailed Description . . . . .	280
8.267.2 Field Documentation . . . . .	281
8.267.2.1 dl_bw_value . . . . .	281
8.267.2.2 TlvPresent . . . . .	281
8.268nas_PhyCaAggScellIndex Struct Reference . . . . .	281
8.268.1 Detailed Description . . . . .	281
8.268.2 Field Documentation . . . . .	281

8.268.2.1 scell_idx . . . . .	281
8.268.2.2 TlvPresent . . . . .	281
8.269nas_PhyCaAggScellIndType Struct Reference . . . . .	281
8.269.1 Detailed Description . . . . .	281
8.269.2 Field Documentation . . . . .	282
8.269.2.1 freq . . . . .	282
8.269.2.2 pci . . . . .	282
8.269.2.3 scell_state . . . . .	282
8.269.2.4 TlvPresent . . . . .	282
8.270nas_PhyCaAggScellInfo Struct Reference . . . . .	282
8.270.1 Detailed Description . . . . .	282
8.270.2 Field Documentation . . . . .	284
8.270.2.1 dl_bw_value . . . . .	284
8.270.2.2 freq . . . . .	284
8.270.2.3 iLTEbandValue . . . . .	284
8.270.2.4 pci . . . . .	284
8.270.2.5 scell_state . . . . .	284
8.270.2.6 TlvPresent . . . . .	284
8.271nas_PilotSetData Struct Reference . . . . .	284
8.271.1 Detailed Description . . . . .	284
8.271.2 Field Documentation . . . . .	284
8.271.2.1 NumPilots . . . . .	284
8.271.2.2 pPilotSetInfo . . . . .	284
8.272nas_PilotSetParams Struct Reference . . . . .	285
8.272.1 Detailed Description . . . . .	285
8.272.2 Field Documentation . . . . .	285
8.272.2.1 PilotPN . . . . .	285
8.272.2.2 PilotStrength . . . . .	285
8.272.2.3 PilotType . . . . .	285
8.273nas_PlmnID Struct Reference . . . . .	285
8.273.1 Detailed Description . . . . .	285
8.273.2 Field Documentation . . . . .	286
8.273.2.1 mcc . . . . .	286
8.273.2.2 mnc . . . . .	286
8.273.2.3 pcsDigit . . . . .	286
8.273.2.4 TlvPresent . . . . .	286
8.274nas_PLMNNetworkName Struct Reference . . . . .	286
8.274.1 Detailed Description . . . . .	286
8.274.2 Field Documentation . . . . .	286
8.274.2.1 numInstance . . . . .	286

8.274.2.2 PLMNNetName . . . . .	286
8.275nas_PLMNNetworkNameData Struct Reference . . . . .	286
8.275.1 Detailed Description . . . . .	287
8.275.2 Field Documentation . . . . .	288
8.275.2.1 codingScheme . . . . .	288
8.275.2.2 countryInitials . . . . .	288
8.275.2.3 longName . . . . .	288
8.275.2.4 longNameLen . . . . .	288
8.275.2.5 longNameSpareBits . . . . .	288
8.275.2.6 shortName . . . . .	288
8.275.2.7 shortNameLen . . . . .	288
8.275.2.8 shortNameSpareBits . . . . .	288
8.276nas_PRLPrefTlv Struct Reference . . . . .	288
8.276.1 Detailed Description . . . . .	288
8.276.2 Field Documentation . . . . .	289
8.276.2.1 PRLPref . . . . .	289
8.276.2.2 TlvPresent . . . . .	289
8.277nas_protocolSubtypeElement Struct Reference . . . . .	289
8.277.1 Detailed Description . . . . .	289
8.277.2 Field Documentation . . . . .	290
8.277.2.1 AccessMac . . . . .	290
8.277.2.2 AuthProt . . . . .	290
8.277.2.3 ControlMac . . . . .	290
8.277.2.4 EncryptProt . . . . .	290
8.277.2.5 ForwardMac . . . . .	290
8.277.2.6 IdleState . . . . .	290
8.277.2.7 KeyExchange . . . . .	290
8.277.2.8 MultDisc . . . . .	290
8.277.2.9 PhysicalLayer . . . . .	290
8.277.2.10ReverseMac . . . . .	290
8.277.2.11SecProt . . . . .	290
8.277.2.12VirtStream . . . . .	290
8.278nas_qaQmi3Gpp2TimeZone Struct Reference . . . . .	290
8.278.1 Detailed Description . . . . .	291
8.278.2 Field Documentation . . . . .	291
8.278.2.1 daylightSavings . . . . .	291
8.278.2.2 leapSeconds . . . . .	291
8.278.2.3 localTimeOffset . . . . .	291
8.279nas_QmiNas3GppNetworkInfo Struct Reference . . . . .	291
8.279.1 Detailed Description . . . . .	291

8.279.2 Field Documentation . . . . .	292
8.279.2.1 Description . . . . .	292
8.279.2.2 Forbidden . . . . .	292
8.279.2.3 InUse . . . . .	292
8.279.2.4 MCC . . . . .	292
8.279.2.5 MNC . . . . .	292
8.279.2.6 Preferred . . . . .	292
8.279.2.7 Roaming . . . . .	292
8.280nas_QmiNas3GppNetworkRAT Struct Reference . . . . .	292
8.280.1 Detailed Description . . . . .	293
8.280.2 Field Documentation . . . . .	293
8.280.2.1 MCC . . . . .	293
8.280.2.2 MNC . . . . .	293
8.280.2.3 RAT . . . . .	293
8.281nas_QmisNasPcsDigit Struct Reference . . . . .	293
8.281.1 Detailed Description . . . . .	293
8.281.2 Field Documentation . . . . .	294
8.281.2.1 includes_pcs_digit . . . . .	294
8.281.2.2 MCC . . . . .	294
8.281.2.3 MNC . . . . .	294
8.282nas_QmisNasSlqsNasPCICellInfo Struct Reference . . . . .	294
8.282.1 Detailed Description . . . . .	294
8.282.2 Field Documentation . . . . .	294
8.282.2.1 cellID . . . . .	294
8.282.2.2 freq . . . . .	294
8.282.2.3 GlobalCellID . . . . .	294
8.282.2.4 nasQmisNasPcsDigit . . . . .	294
8.282.2.5 PlmnLen . . . . .	294
8.283nas_QmisNasSlqsNasPCIInfo Struct Reference . . . . .	295
8.283.1 Detailed Description . . . . .	295
8.283.2 Field Documentation . . . . .	295
8.283.2.1 nasQmisNasSlqsNasPCICellInfo . . . . .	295
8.283.2.2 PCICellInfoLen . . . . .	295
8.283.2.3 rsrp . . . . .	295
8.283.2.4 rsrpRx0 . . . . .	295
8.283.2.5 rsrpRx1 . . . . .	296
8.283.2.6 rsrq . . . . .	296
8.283.2.7 rsrqRx0 . . . . .	296
8.283.2.8 rsrqRx1 . . . . .	296
8.284nas_RankIndicatorTlv Struct Reference . . . . .	296

8.284.1 Detailed Description . . . . .	296
8.284.2 Field Documentation . . . . .	296
8.284.2.1 count0 . . . . .	296
8.284.2.2 count1 . . . . .	296
8.284.2.3 TlvPresent . . . . .	296
8.285nas_RatDisabledMaskTlv Struct Reference . . . . .	296
8.285.1 Detailed Description . . . . .	296
8.285.2 Field Documentation . . . . .	297
8.285.2.1 ratDisabledMask . . . . .	297
8.285.2.2 TlvPresent . . . . .	297
8.286nas_RejectReasonTlv Struct Reference . . . . .	297
8.286.1 Detailed Description . . . . .	297
8.286.2 Field Documentation . . . . .	297
8.286.2.1 rejectCause . . . . .	297
8.286.2.2 serviceDomain . . . . .	297
8.286.2.3 TlvPresent . . . . .	297
8.287nas_RfBandInfoElements Struct Reference . . . . .	297
8.287.1 Detailed Description . . . . .	298
8.287.2 Field Documentation . . . . .	298
8.287.2.1 activeBandClass . . . . .	298
8.287.2.2 activeChannel . . . . .	298
8.287.2.3 radiolInterface . . . . .	298
8.288nas_RfBandInfoExtFormat Struct Reference . . . . .	298
8.288.1 Detailed Description . . . . .	298
8.288.2 Field Documentation . . . . .	299
8.288.2.1 plnstancesSize . . . . .	299
8.288.2.2 pRfBandInfoExtFormatParam . . . . .	299
8.288.2.3 TlvPresent . . . . .	299
8.289nas_RfBandInfoExtFormatElements Struct Reference . . . . .	299
8.289.1 Detailed Description . . . . .	299
8.289.2 Field Documentation . . . . .	299
8.289.2.1 activeBand . . . . .	299
8.289.2.2 activeChannel . . . . .	299
8.289.2.3 radiolInterface . . . . .	300
8.290nas_RfBandInfoExtTlv Struct Reference . . . . .	300
8.290.1 Detailed Description . . . . .	300
8.290.2 Field Documentation . . . . .	300
8.290.2.1 activeBand . . . . .	301
8.290.2.2 activeChannel . . . . .	301
8.290.2.3 radiolf . . . . .	301

8.290.2.4 TlvPresent . . . . .	301
8.291nas_RfBandInfoList Struct Reference . . . . .	301
8.291.1 Detailed Description . . . . .	301
8.291.2 Field Documentation . . . . .	301
8.291.2.1 pInstanceSize . . . . .	301
8.291.2.2 pRFBandInfoParam . . . . .	301
8.292nas_RFBandInfoTlv Struct Reference . . . . .	301
8.292.1 Detailed Description . . . . .	301
8.292.2 Field Documentation . . . . .	302
8.292.2.1 activeBand . . . . .	302
8.292.2.2 activeChannel . . . . .	302
8.292.2.3 radiolf . . . . .	302
8.292.2.4 TlvPresent . . . . .	302
8.293nas_RfBandwidthInfo Struct Reference . . . . .	302
8.293.1 Detailed Description . . . . .	303
8.293.2 Field Documentation . . . . .	303
8.293.2.1 pInstancesSize . . . . .	303
8.293.2.2 pRfBandwidthInfoParam . . . . .	303
8.293.2.3 TlvPresent . . . . .	303
8.294nas_RfBandwidthInfoElements Struct Reference . . . . .	303
8.294.1 Detailed Description . . . . .	303
8.294.2 Field Documentation . . . . .	304
8.294.2.1 bandwidth . . . . .	304
8.294.2.2 radiolInterface . . . . .	304
8.295nas_RFBandwidthInfoTlv Struct Reference . . . . .	304
8.295.1 Detailed Description . . . . .	304
8.295.2 Field Documentation . . . . .	305
8.295.2.1 bandwidth . . . . .	305
8.295.2.2 radiolf . . . . .	305
8.295.2.3 TlvPresent . . . . .	305
8.296nas_RfDedicatedBandInfo Struct Reference . . . . .	305
8.296.1 Detailed Description . . . . .	306
8.296.2 Field Documentation . . . . .	306
8.296.2.1 pInstancesSize . . . . .	306
8.296.2.2 pRfDedicatedBandInfoParam . . . . .	306
8.296.2.3 TlvPresent . . . . .	306
8.297nas_RfDedicatedBandInfoElements Struct Reference . . . . .	306
8.297.1 Detailed Description . . . . .	306
8.297.2 Field Documentation . . . . .	307
8.297.2.1 dedicatedBand . . . . .	307

8.297.2.2 radiolInterface . . . . .	307
8.298nas_RFDedicatedBandInfoTlv Struct Reference . . . . .	307
8.298.1 Detailed Description . . . . .	307
8.298.2 Field Documentation . . . . .	307
8.298.2.1 dedicatedBand . . . . .	307
8.298.2.2 radiolf . . . . .	308
8.298.2.3 TlvPresent . . . . .	308
8.299nas_RFInfoTlv Struct Reference . . . . .	308
8.299.1 Detailed Description . . . . .	308
8.299.2 Field Documentation . . . . .	308
8.299.2.1 activeBandClass . . . . .	308
8.299.2.2 activeChannel . . . . .	308
8.299.2.3 radiolInterface . . . . .	309
8.299.2.4 radiolInterfaceSize . . . . .	309
8.299.2.5 TlvPresent . . . . .	309
8.300nas_roamIndList Struct Reference . . . . .	309
8.300.1 Detailed Description . . . . .	309
8.300.2 Field Documentation . . . . .	309
8.300.2.1 numInstances . . . . .	309
8.300.2.2 radiolInterface . . . . .	309
8.300.2.3 roamIndicator . . . . .	309
8.301nas_RoamPrefTlv Struct Reference . . . . .	310
8.301.1 Detailed Description . . . . .	310
8.301.2 Field Documentation . . . . .	310
8.301.2.1 RoamPref . . . . .	310
8.301.2.2 TlvPresent . . . . .	310
8.302nas_RSRPThresh Struct Reference . . . . .	310
8.302.1 Detailed Description . . . . .	310
8.302.2 Field Documentation . . . . .	311
8.302.2.1 pRSRPThresList . . . . .	311
8.302.2.2 RSRPThresListLen . . . . .	311
8.303nas_rsrqInformation Struct Reference . . . . .	311
8.303.1 Detailed Description . . . . .	311
8.303.2 Field Documentation . . . . .	311
8.303.2.1 radiolf . . . . .	311
8.303.2.2 rsrq . . . . .	311
8.304nas_RSRQThresh Struct Reference . . . . .	311
8.304.1 Detailed Description . . . . .	312
8.304.2 Field Documentation . . . . .	312
8.304.2.1 pRSRQThresList . . . . .	312



8.304.2.2 RSRQThresListLen . . . . .	312
8.305nas_RSSIThresh Struct Reference . . . . .	312
8.305.1 Detailed Description . . . . .	312
8.305.2 Field Documentation . . . . .	313
8.305.2.1 pRSSIThresList . . . . .	313
8.305.2.2 RSSIThresListLen . . . . .	313
8.306nas_rxInfo Struct Reference . . . . .	313
8.306.1 Detailed Description . . . . .	313
8.306.2 Field Documentation . . . . .	314
8.306.2.1 ecio . . . . .	314
8.306.2.2 isRadioTuned . . . . .	314
8.306.2.3 phase . . . . .	314
8.306.2.4 rscp . . . . .	314
8.306.2.5 rsrp . . . . .	314
8.306.2.6 rxPower . . . . .	314
8.307nas_RxSigInfo Struct Reference . . . . .	314
8.307.1 Detailed Description . . . . .	314
8.307.2 Field Documentation . . . . .	315
8.307.2.1 isRadioTuned . . . . .	315
8.307.2.2 rsrp . . . . .	315
8.307.2.3 rxChainIndex . . . . .	315
8.307.2.4 rxPower . . . . .	315
8.308nas_rxSignalStrengthListElement Struct Reference . . . . .	315
8.308.1 Detailed Description . . . . .	315
8.308.2 Field Documentation . . . . .	316
8.308.2.1 radiolf . . . . .	316
8.308.2.2 rxSignalStrength . . . . .	316
8.309nas_SccRxInfo Struct Reference . . . . .	316
8.309.1 Detailed Description . . . . .	316
8.309.2 Field Documentation . . . . .	317
8.309.2.1 numInstances . . . . .	317
8.309.2.2 rsrq . . . . .	317
8.309.2.3 sigInfo . . . . .	317
8.309.2.4 snr . . . . .	317
8.309.2.5 TlvPresent . . . . .	317
8.310nas_serviceProviderName Struct Reference . . . . .	317
8.310.1 Detailed Description . . . . .	317
8.310.2 Field Documentation . . . . .	317
8.310.2.1 displayCondition . . . . .	317
8.310.2.2 spn . . . . .	317

8.310.2.3 spnLength . . . . .	317
8.311nas_servSystem Struct Reference . . . . .	318
8.311.1 Detailed Description . . . . .	318
8.311.2 Field Documentation . . . . .	319
8.311.2.1 csAttachState . . . . .	319
8.311.2.2 numRadioInterfaces . . . . .	319
8.311.2.3 psAttachState . . . . .	319
8.311.2.4 radiolInterface . . . . .	319
8.311.2.5 regState . . . . .	319
8.311.2.6 selNetwork . . . . .	319
8.312nas_sidNid Struct Reference . . . . .	319
8.312.1 Detailed Description . . . . .	319
8.312.2 Field Documentation . . . . .	319
8.312.2.1 nid . . . . .	319
8.312.2.2 sid . . . . .	319
8.313nas_SignalStrengthTlv Struct Reference . . . . .	320
8.313.1 Detailed Description . . . . .	320
8.313.2 Field Documentation . . . . .	320
8.313.2.1 radiolInterface . . . . .	320
8.313.2.2 signalStrength . . . . .	320
8.313.2.3 TlvPresent . . . . .	320
8.314nas_SimRejInfoTlv Struct Reference . . . . .	320
8.314.1 Detailed Description . . . . .	320
8.314.2 Field Documentation . . . . .	321
8.314.2.1 SimRejInfo . . . . .	321
8.314.2.2 TLVPresent . . . . .	321
8.315nas_SLQSSignalStrengthsIndReq Struct Reference . . . . .	321
8.315.1 Detailed Description . . . . .	321
8.315.2 Field Documentation . . . . .	322
8.315.2.1 ecioDelta . . . . .	322
8.315.2.2 ecioThresholdList . . . . .	322
8.315.2.3 ecioThresholdListLen . . . . .	322
8.315.2.4 ioDelta . . . . .	322
8.315.2.5 lteRsrpDelta . . . . .	322
8.315.2.6 lteSnrDelta . . . . .	322
8.315.2.7 rsrqDelta . . . . .	322
8.315.2.8 rxSignalStrengthDelta . . . . .	322
8.315.2.9 sinrDelta . . . . .	322
8.315.2.10sinrThresholdList . . . . .	322
8.315.2.11sinrThresholdListLen . . . . .	322

8.316nas_SLQSSignalStrengthsInformation Struct Reference	322
8.316.1 Detailed Description	323
8.316.2 Field Documentation	323
8.316.2.1 eciInfo	323
8.316.2.2 errorRateInfo	324
8.316.2.3 io	324
8.316.2.4 lteRsrpinfo	324
8.316.2.5 lteSnrinfo	324
8.316.2.6 rsrqInfo	324
8.316.2.7 rxSignalStrengthInfo	324
8.316.2.8 sinr	324
8.317nas_SLQSSignalStrengthsTlv Struct Reference	324
8.317.1 Detailed Description	324
8.317.2 Field Documentation	324
8.317.2.1 sSLQSSignalStrengthsInfo	324
8.317.2.2 TlvPresent	324
8.318nas_SrvDomainPrefTlv Struct Reference	324
8.318.1 Detailed Description	324
8.318.2 Field Documentation	325
8.318.2.1 SrvDomainPref	325
8.318.2.2 TlvPresent	325
8.319nas_SrvRegRestrictionTlv Struct Reference	325
8.319.1 Detailed Description	325
8.319.2 Field Documentation	325
8.319.2.1 SrvRegRestriction	325
8.319.2.2 TLVPresent	325
8.320nas_SrvRegRestrictTlv Struct Reference	326
8.320.1 Detailed Description	326
8.320.2 Field Documentation	326
8.320.2.1 SrvRegRestriction	326
8.320.2.2 TlvPresent	326
8.321nas_SrvStatusInfo Struct Reference	326
8.321.1 Detailed Description	326
8.321.2 Field Documentation	327
8.321.2.1 isPrefDataPath	327
8.321.2.2 srvStatus	327
8.322nas_sysInfoCommon Struct Reference	327
8.322.1 Detailed Description	327
8.322.2 Field Documentation	329
8.322.2.1 isSysForbidden	329

8.322.2.2 isSysForbiddenValid . . . . .	329
8.322.2.3 roamStatus . . . . .	329
8.322.2.4 roamStatusValid . . . . .	329
8.322.2.5 srvCapability . . . . .	329
8.322.2.6 srvCapabilityValid . . . . .	329
8.322.2.7 srvDomain . . . . .	329
8.322.2.8 srvDomainValid . . . . .	329
8.323nas_TDSCDMABandPrefTlv Struct Reference . . . . .	329
8.323.1 Detailed Description . . . . .	329
8.323.2 Field Documentation . . . . .	330
8.323.2.1 TdscdmaBandPref . . . . .	330
8.323.2.2 TlvPresent . . . . .	330
8.324nas_TDSCDMAECIOThresh Struct Reference . . . . .	330
8.324.1 Detailed Description . . . . .	330
8.324.2 Field Documentation . . . . .	330
8.324.2.1 pTDSCDMAECIOThreshList . . . . .	330
8.324.2.2 TDSCDMAECIOThreshListLen . . . . .	330
8.325nas_TDSCDMARSCPThresh Struct Reference . . . . .	330
8.325.1 Detailed Description . . . . .	330
8.325.2 Field Documentation . . . . .	331
8.325.2.1 pTDSCDMARSCPThreshList . . . . .	331
8.325.2.2 TDSCDMARSCPThreshListLen . . . . .	331
8.326nas_TDSCDMARSSIThresh Struct Reference . . . . .	331
8.326.1 Detailed Description . . . . .	331
8.326.2 Field Documentation . . . . .	331
8.326.2.1 pTDSCDMARSSIThreshList . . . . .	331
8.326.2.2 TDSCDMARSSIThreshListLen . . . . .	331
8.327nas_TDSCDMASINRCONFThresh Struct Reference . . . . .	331
8.327.1 Detailed Description . . . . .	331
8.327.2 Field Documentation . . . . .	332
8.327.2.1 pTDSCDMASINRCONFThreshList . . . . .	332
8.327.2.2 TDSCDMASINRCONFThreshListLen . . . . .	332
8.328nas_TDSCDMASINRThresh Struct Reference . . . . .	332
8.328.1 Detailed Description . . . . .	332
8.328.2 Field Documentation . . . . .	332
8.328.2.1 pTDSCDMASINRThreshList . . . . .	332
8.328.2.2 TDSCDMASINRThreshListLen . . . . .	332
8.329nas_timeInfo Struct Reference . . . . .	332
8.329.1 Detailed Description . . . . .	332
8.329.2 Field Documentation . . . . .	334

8.329.2.1 day . . . . .	334
8.329.2.2 dayLtSavingAdj . . . . .	334
8.329.2.3 dayOfWeek . . . . .	334
8.329.2.4 hour . . . . .	334
8.329.2.5 minute . . . . .	334
8.329.2.6 month . . . . .	334
8.329.2.7 radiolInterface . . . . .	334
8.329.2.8 second . . . . .	334
8.329.2.9 timeZone . . . . .	334
8.329.2.10TlvPresent . . . . .	334
8.329.2.11year . . . . .	334
8.330nas_trueIMSI Struct Reference . . . . .	334
8.330.1 Detailed Description . . . . .	334
8.330.2 Field Documentation . . . . .	335
8.330.2.1 imsiT1112 . . . . .	335
8.330.2.2 imsiTaddrNum . . . . .	335
8.330.2.3 imsiTS1 . . . . .	335
8.330.2.4 imsiTS2 . . . . .	335
8.330.2.5 mccT . . . . .	335
8.331nas_txInfo Struct Reference . . . . .	335
8.331.1 Detailed Description . . . . .	335
8.331.2 Field Documentation . . . . .	335
8.331.2.1 isInTraffic . . . . .	335
8.331.2.2 txPower . . . . .	335
8.332nas_UMTSExtInfo Struct Reference . . . . .	335
8.332.1 Detailed Description . . . . .	336
8.332.2 Field Documentation . . . . .	337
8.332.2.1 cellId . . . . .	337
8.332.2.2 ecio . . . . .	337
8.332.2.3 geranInst . . . . .	337
8.332.2.4 geranInstElement . . . . .	337
8.332.2.5 lac . . . . .	337
8.332.2.6 plmn . . . . .	337
8.332.2.7 psc . . . . .	337
8.332.2.8 rscp . . . . .	337
8.332.2.9 squal . . . . .	337
8.332.2.10srxFwd . . . . .	337
8.332.2.11TlvPresent . . . . .	337
8.332.2.12uarfcn . . . . .	337
8.332.2.13umtsInst . . . . .	337

8.332.2.14umtsInstElement . . . . .	337
8.333nas_UMTSInfo Struct Reference . . . . .	337
8.333.1 Detailed Description . . . . .	338
8.333.2 Field Documentation . . . . .	339
8.333.2.1 cellID . . . . .	339
8.333.2.2 ecio . . . . .	339
8.333.2.3 geranInst . . . . .	339
8.333.2.4 GeranInstInfo . . . . .	339
8.333.2.5 lac . . . . .	339
8.333.2.6 plmn . . . . .	339
8.333.2.7 psc . . . . .	339
8.333.2.8 rscp . . . . .	339
8.333.2.9 uarfcn . . . . .	339
8.333.2.10umtsInst . . . . .	339
8.333.2.11UMTSInstInfo . . . . .	339
8.334nas_umtsInstArr Struct Reference . . . . .	339
8.334.1 Detailed Description . . . . .	339
8.334.2 Field Documentation . . . . .	340
8.334.2.1 umtsEcio . . . . .	340
8.334.2.2 umtsPsc . . . . .	340
8.334.2.3 umtsRank . . . . .	340
8.334.2.4 umtsRscp . . . . .	340
8.334.2.5 umtsSet . . . . .	340
8.334.2.6 umtsSqual . . . . .	340
8.334.2.7 umtsSrxLvl . . . . .	340
8.334.2.8 umtsUarfcn . . . . .	340
8.335nas_UMTSinstInfo Struct Reference . . . . .	340
8.335.1 Detailed Description . . . . .	341
8.335.2 Field Documentation . . . . .	341
8.335.2.1 umtsEcio . . . . .	341
8.335.2.2 umtsPsc . . . . .	341
8.335.2.3 umtsRscp . . . . .	341
8.335.2.4 umtsUarfcn . . . . .	341
8.336nas_umtsLTENbrCell Struct Reference . . . . .	341
8.336.1 Detailed Description . . . . .	341
8.336.2 Field Documentation . . . . .	342
8.336.2.1 cellIsTDD . . . . .	342
8.336.2.2 earfcn . . . . .	342
8.336.2.3 pci . . . . .	342
8.336.2.4 rsrp . . . . .	342

8.336.2.5 rsrq . . . . .	342
8.336.2.6 srxlev . . . . .	342
8.337nas_UniversalTime Struct Reference . . . . .	342
8.337.1 Detailed Description . . . . .	342
8.337.2 Field Documentation . . . . .	343
8.337.2.1 day . . . . .	343
8.337.2.2 dayOfWeek . . . . .	343
8.337.2.3 hour . . . . .	343
8.337.2.4 minute . . . . .	343
8.337.2.5 month . . . . .	343
8.337.2.6 second . . . . .	343
8.337.2.7 year . . . . .	343
8.338nas_UsageSettingTlv Struct Reference . . . . .	343
8.338.1 Detailed Description . . . . .	343
8.338.2 Field Documentation . . . . .	344
8.338.2.1 TlvPresent . . . . .	344
8.338.2.2 UsageSetting . . . . .	344
8.339nas_VoiceDomainPrefTlv Struct Reference . . . . .	344
8.339.1 Detailed Description . . . . .	344
8.339.2 Field Documentation . . . . .	344
8.339.2.1 TlvPresent . . . . .	344
8.339.2.2 VoiceDomainPref . . . . .	344
8.340nas_wcdmaCellInfo Struct Reference . . . . .	344
8.340.1 Detailed Description . . . . .	345
8.340.2 Field Documentation . . . . .	345
8.340.2.1 cpich_ecno . . . . .	345
8.340.2.2 cpich_rscp . . . . .	345
8.340.2.3 psc . . . . .	345
8.340.2.4 srxlev . . . . .	345
8.341nas_WCDMACellInfoExt Struct Reference . . . . .	345
8.341.1 Detailed Description . . . . .	346
8.341.2 Field Documentation . . . . .	346
8.341.2.1 TlvPresent . . . . .	346
8.341.2.2 wAgc . . . . .	346
8.341.2.3 wDIBler . . . . .	346
8.341.2.4 wTxAgc . . . . .	346
8.342nas_WCDMAECIOThresh Struct Reference . . . . .	346
8.342.1 Detailed Description . . . . .	346
8.342.2 Field Documentation . . . . .	346
8.342.2.1 pWCDMAECIOThreshList . . . . .	347

8.342.2.2 WCDMAECIOThreshListLen . . . . .	347
8.343nas_WCDMAInfoLTENeighborCell Struct Reference . . . . .	347
8.343.1 Detailed Description . . . . .	347
8.343.2 Field Documentation . . . . .	347
8.343.2.1 UMTSLTENbrCell . . . . .	347
8.343.2.2 umtsLTENbrCellLen . . . . .	347
8.343.2.3 wcdmaRRCState . . . . .	347
8.344nas_WCDMARSSIThresh Struct Reference . . . . .	347
8.344.1 Detailed Description . . . . .	348
8.344.2 Field Documentation . . . . .	348
8.344.2.1 pWCDMARSSIThreshList . . . . .	348
8.344.2.2 WCDMARSSIThreshListLen . . . . .	348
8.345nas_WCDMASysInfo Struct Reference . . . . .	348
8.345.1 Detailed Description . . . . .	348
8.345.2 Field Documentation . . . . .	351
8.345.2.1 cellId . . . . .	351
8.345.2.2 cellIdValid . . . . .	351
8.345.2.3 hsCallStatus . . . . .	351
8.345.2.4 hsCallStatusValid . . . . .	351
8.345.2.5 hsInd . . . . .	351
8.345.2.6 hsIndValid . . . . .	351
8.345.2.7 lac . . . . .	351
8.345.2.8 lacValid . . . . .	351
8.345.2.9 MCC . . . . .	351
8.345.2.10MNC . . . . .	351
8.345.2.11networkIdValid . . . . .	351
8.345.2.12psc . . . . .	351
8.345.2.13pscValid . . . . .	351
8.345.2.14regRejectInfoValid . . . . .	351
8.345.2.15rejCause . . . . .	351
8.345.2.16rejectSrvDomain . . . . .	351
8.345.2.17sysInfoWCDMA . . . . .	351
8.346nas_wcdmaUARFCN Struct Reference . . . . .	351
8.346.1 Detailed Description . . . . .	352
8.346.2 Field Documentation . . . . .	352
8.346.2.1 status . . . . .	352
8.346.2.2 uarfcn . . . . .	352
8.347NASAcqOrderPrefTlv Struct Reference . . . . .	352
8.347.1 Detailed Description . . . . .	352
8.347.2 Field Documentation . . . . .	352



8.347.2.1 AcqOrderLen . . . . .	352
8.347.2.2 AcqOrderPref . . . . .	352
8.347.2.3 TlvPresent . . . . .	352
8.348NASBandPreferenceTlv Struct Reference . . . . .	352
8.348.1 Detailed Description . . . . .	353
8.348.2 Field Documentation . . . . .	353
8.348.2.1 band_pref . . . . .	353
8.348.2.2 TlvPresent . . . . .	353
8.349NASCiotAcqOrderPrefTlv Struct Reference . . . . .	353
8.349.1 Detailed Description . . . . .	353
8.349.2 Field Documentation . . . . .	353
8.349.2.1 CiotAcqOrderLen . . . . .	353
8.349.2.2 CiotAcqOrderPref . . . . .	353
8.349.2.3 TlvPresent . . . . .	353
8.350NASCiotLteOpModePrefTlv Struct Reference . . . . .	353
8.350.1 Detailed Description . . . . .	353
8.350.2 Field Documentation . . . . .	354
8.350.2.1 CiotLteOpModePref . . . . .	354
8.350.2.2 TlvPresent . . . . .	354
8.351NASEmergencyModeTlv Struct Reference . . . . .	354
8.351.1 Detailed Description . . . . .	354
8.351.2 Field Documentation . . . . .	354
8.351.2.1 EmerMode . . . . .	354
8.351.2.2 TlvPresent . . . . .	354
8.352NasGetLTECphyCalInfo Struct Reference . . . . .	354
8.352.1 Detailed Description . . . . .	354
8.352.2 Field Documentation . . . . .	355
8.352.2.1 PhyCaAggPcellInfo . . . . .	355
8.352.2.2 PhyCaAggScellArray . . . . .	355
8.352.2.3 PhyCaAggScellIDBw . . . . .	355
8.352.2.4 PhyCaAggScellIndex . . . . .	355
8.352.2.5 PhyCaAggScellIndType . . . . .	355
8.352.2.6 PhyCaAggScellInfo . . . . .	355
8.353NASGWAcqOrderPrefTlv Struct Reference . . . . .	355
8.353.1 Detailed Description . . . . .	355
8.353.2 Field Documentation . . . . .	356
8.353.2.1 GWAcqOrderPref . . . . .	356
8.353.2.2 TlvPresent . . . . .	356
8.354NASLTEBandPreferenceTlv Struct Reference . . . . .	356
8.354.1 Detailed Description . . . . .	356

8.354.2 Field Documentation . . . . .	356
8.354.2.1 LTEBandPref . . . . .	356
8.354.2.2 TlvPresent . . . . .	356
8.355NASLteM1BandPrefTlv Struct Reference . . . . .	356
8.355.1 Detailed Description . . . . .	356
8.355.2 Field Documentation . . . . .	356
8.355.2.1 LteM1BandPref . . . . .	357
8.355.2.2 TlvPresent . . . . .	357
8.356NASLteNasReleaseInfoTlv Struct Reference . . . . .	357
8.356.1 Detailed Description . . . . .	357
8.356.2 Field Documentation . . . . .	357
8.356.2.1 nas_major . . . . .	357
8.356.2.2 nas_minor . . . . .	357
8.356.2.3 nas_release . . . . .	357
8.356.2.4 TlvPresent . . . . .	357
8.357NASLteNB1BandPrefTlv Struct Reference . . . . .	357
8.357.1 Detailed Description . . . . .	357
8.357.2 Field Documentation . . . . .	358
8.357.2.1 LteNB1BandPref . . . . .	358
8.357.2.2 TlvPresent . . . . .	358
8.358NASModePreferenceTlv Struct Reference . . . . .	358
8.358.1 Detailed Description . . . . .	358
8.358.2 Field Documentation . . . . .	358
8.358.2.1 ModePref . . . . .	358
8.358.2.2 TlvPresent . . . . .	358
8.359NASNetSelPreferenceTlv Struct Reference . . . . .	358
8.359.1 Detailed Description . . . . .	358
8.359.2 Field Documentation . . . . .	358
8.359.2.1 NetSelPref . . . . .	358
8.359.2.2 TlvPresent . . . . .	359
8.360NASNr5gBandPrefTlv Struct Reference . . . . .	359
8.360.1 Detailed Description . . . . .	359
8.360.2 Field Documentation . . . . .	359
8.360.2.1 Nr5gBandPrefbits1_64 . . . . .	359
8.360.2.2 Nr5gBandPrefbits_129_192 . . . . .	359
8.360.2.3 Nr5gBandPrefbits_193_256 . . . . .	359
8.360.2.4 Nr5gBandPrefbits_65_128 . . . . .	359
8.360.2.5 TlvPresent . . . . .	359
8.361NASOTAMessageTlv Struct Reference . . . . .	359
8.361.1 Detailed Description . . . . .	359

8.361.2 Field Documentation	360
8.361.2.1 data_buf	360
8.361.2.2 data_len	360
8.361.2.3 message_type	360
8.361.2.4 TlvPresent	360
8.362NASPhyCaAggPcellInfo Struct Reference	360
8.362.1 Detailed Description	360
8.362.2 Field Documentation	361
8.362.2.1 dl_bw_value	361
8.362.2.2 freq	361
8.362.2.3 iLTEbandValue	361
8.362.2.4 pci	361
8.362.2.5 TlvPresent	361
8.363NASPhyCaAggScellArray Struct Reference	361
8.363.1 Detailed Description	361
8.363.2 Field Documentation	362
8.363.2.1 band	362
8.363.2.2 cphy_ca_dl_bandwidth	362
8.363.2.3 cphy_scell_info_list_len	362
8.363.2.4 freq	362
8.363.2.5 pci	362
8.363.2.6 scell_idx	362
8.363.2.7 scell_state	362
8.363.2.8 TlvPresent	362
8.364NASPhyCaAggScellIDBw Struct Reference	362
8.364.1 Detailed Description	363
8.364.2 Field Documentation	363
8.364.2.1 dl_bw_value	363
8.364.2.2 TlvPresent	363
8.365NASPhyCaAggScellIndex Struct Reference	363
8.365.1 Detailed Description	363
8.365.2 Field Documentation	363
8.365.2.1 scell_idx	363
8.365.2.2 TlvPresent	363
8.366NASPhyCaAggScellIndType Struct Reference	363
8.366.1 Detailed Description	364
8.366.2 Field Documentation	364
8.366.2.1 freq	364
8.366.2.2 pci	364
8.366.2.3 scell_state	364

8.366.2.4 TlvPresent . . . . .	364
8.367NASPhyCaAggScellInfo Struct Reference . . . . .	364
8.367.1 Detailed Description . . . . .	364
8.367.2 Field Documentation . . . . .	365
8.367.2.1 dl_bw_value . . . . .	365
8.367.2.2 freq . . . . .	365
8.367.2.3 iLTEbandValue . . . . .	365
8.367.2.4 pci . . . . .	365
8.367.2.5 scell_state . . . . .	365
8.367.2.6 TlvPresent . . . . .	365
8.368NASPRLPreferenceTlv Struct Reference . . . . .	365
8.368.1 Detailed Description . . . . .	365
8.368.2 Field Documentation . . . . .	365
8.368.2.1 PRLPref . . . . .	366
8.368.2.2 TlvPresent . . . . .	366
8.369NASQmiCbkNasSwtOTAMessageInd Struct Reference . . . . .	366
8.369.1 Detailed Description . . . . .	366
8.369.2 Field Documentation . . . . .	366
8.369.2.1 nasRelInfoTlv . . . . .	366
8.369.2.2 otaMsgTlv . . . . .	366
8.369.2.3 timeTlv . . . . .	366
8.370NASQmiCbkNasSystemSelPrefInd Struct Reference . . . . .	366
8.370.1 Detailed Description . . . . .	367
8.370.2 Field Documentation . . . . .	368
8.370.2.1 AOPTlv . . . . .	368
8.370.2.2 BPTlv . . . . .	368
8.370.2.3 CiotAOPTlv . . . . .	368
8.370.2.4 CiotOpMPTlv . . . . .	368
8.370.2.5 EMTlv . . . . .	368
8.370.2.6 GWAOPTlv . . . . .	368
8.370.2.7 LBPTlv . . . . .	368
8.370.2.8 M1BandPTlv . . . . .	368
8.370.2.9 MPTlv . . . . .	368
8.370.2.10NB1BandPTlv . . . . .	368
8.370.2.11NR5gBandPTlv . . . . .	368
8.370.2.12NSPTlv . . . . .	368
8.370.2.13PRLPTlv . . . . .	369
8.370.2.14RatDMTlv . . . . .	369
8.370.2.15RPTlv . . . . .	369
8.370.2.16SDPTlv . . . . .	369

8.371NASRatDisabledMaskTlv Struct Reference . . . . .	369
8.371.1 Detailed Description . . . . .	369
8.371.2 Field Documentation . . . . .	369
8.371.2.1 RatDisabledMask . . . . .	369
8.371.2.2 TlvPresent . . . . .	369
8.372NASRoamPreferenceTlv Struct Reference . . . . .	369
8.372.1 Detailed Description . . . . .	369
8.372.2 Field Documentation . . . . .	369
8.372.2.1 RoamPref . . . . .	369
8.372.2.2 TlvPresent . . . . .	369
8.373NASServDomainPrefTlv Struct Reference . . . . .	370
8.373.1 Detailed Description . . . . .	370
8.373.2 Field Documentation . . . . .	370
8.373.2.1 SrvDomainPref . . . . .	370
8.373.2.2 TlvPresent . . . . .	370
8.374NASServingSystemInfo Struct Reference . . . . .	370
8.374.1 Detailed Description . . . . .	370
8.374.2 Field Documentation . . . . .	371
8.374.2.1 csAttachState . . . . .	371
8.374.2.2 hdrPersonality . . . . .	371
8.374.2.3 psAttachState . . . . .	371
8.374.2.4 radiolInterfaceList . . . . .	371
8.374.2.5 radiolInterfaceNo . . . . .	371
8.374.2.6 registrationState . . . . .	371
8.374.2.7 selectedNetwork . . . . .	371
8.375NASTimeInfoTlv Struct Reference . . . . .	371
8.375.1 Detailed Description . . . . .	372
8.375.2 Field Documentation . . . . .	372
8.375.2.1 time . . . . .	372
8.375.2.2 TlvPresent . . . . .	372
8.376newMTMessageTlv Struct Reference . . . . .	372
8.376.1 Detailed Description . . . . .	372
8.376.2 Field Documentation . . . . .	372
8.376.2.1 MTMessageInfo . . . . .	372
8.376.2.2 TlvPresent . . . . .	372
8.377pack_audio_SLQSGetAudioPathConfig_t Struct Reference . . . . .	372
8.377.1 Detailed Description . . . . .	373
8.377.2 Field Documentation . . . . .	373
8.377.2.1 Item . . . . .	373
8.377.2.2 Profile . . . . .	373

8.378	<a href="#">pack_audio_SLQSGetAudioProfile_t Struct Reference</a>	373
8.378.1	Detailed Description	373
8.378.2	Field Documentation	374
8.378.2.1	Generator	374
8.379	<a href="#">pack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference</a>	374
8.379.1	Detailed Description	374
8.379.2	Field Documentation	374
8.379.2.1	Generator	374
8.379.2.2	Item	374
8.379.2.3	Profile	374
8.379.2.4	Volume	374
8.380	<a href="#">pack_audio_SLQSSetAudioPathConfig_t Struct Reference</a>	374
8.380.1	Detailed Description	375
8.380.2	Field Documentation	376
8.380.2.1	pCodecSTGain	376
8.380.2.2	pDTMFTXGain	376
8.380.2.3	pECMode	376
8.380.2.4	pNSEnable	376
8.380.2.5	Profile	376
8.380.2.6	pRXAGCList	376
8.380.2.7	pRXAVCAGCSwitch	376
8.380.2.8	pRXAVCList	376
8.380.2.9	pRXPCMIIRFitr	376
8.380.2.10	pTXAGCList	376
8.380.2.11	pTXAVCSwitch	376
8.380.2.12	pTXGain	376
8.380.2.13	pTXPCMIIRFitr	376
8.381	<a href="#">pack_audio_SLQSSetAudioProfile_t Struct Reference</a>	376
8.381.1	Detailed Description	377
8.381.2	Field Documentation	377
8.381.2.1	EarMute	377
8.381.2.2	Generator	377
8.381.2.3	MicMute	377
8.381.2.4	Profile	377
8.381.2.5	Volume	377
8.382	<a href="#">pack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference</a>	378
8.382.1	Detailed Description	378
8.382.2	Field Documentation	378
8.382.2.1	Generator	378
8.382.2.2	Item	378

8.382.2.3 Profile . . . . .	378
8.382.2.4 Volume . . . . .	378
8.382.2.5 VolValue . . . . .	378
8.383pack_cat_CATSendEnvelopeCommand_t Struct Reference . . . . .	378
8.383.1 Detailed Description . . . . .	379
8.383.2 Field Documentation . . . . .	379
8.383.2.1 cmdID . . . . .	379
8.383.2.2 dataLen . . . . .	379
8.383.2.3 pData . . . . .	379
8.384pack_cat_CATSendTerminalResponse_t Struct Reference . . . . .	379
8.384.1 Detailed Description . . . . .	379
8.384.2 Field Documentation . . . . .	380
8.384.2.1 dataLen . . . . .	380
8.384.2.2 pData . . . . .	380
8.384.2.3 refID . . . . .	380
8.385pack_cat_SetCATEventCallback_t Struct Reference . . . . .	380
8.385.1 Detailed Description . . . . .	380
8.385.2 Field Documentation . . . . .	380
8.385.2.1 eventMask . . . . .	380
8.386pack_dms_ActivateAutomatic_t Struct Reference . . . . .	381
8.386.1 Detailed Description . . . . .	381
8.386.2 Field Documentation . . . . .	381
8.386.2.1 actCode . . . . .	381
8.387pack_dms_GetCustFeaturesV2_t Struct Reference . . . . .	381
8.387.1 Detailed Description . . . . .	381
8.387.2 Field Documentation . . . . .	381
8.387.2.1 cust_id . . . . .	381
8.387.2.2 list_type . . . . .	381
8.387.2.3 Tlvresult . . . . .	382
8.388pack_dms_ResetToFactoryDefaults_t Struct Reference . . . . .	382
8.388.1 Detailed Description . . . . .	382
8.388.2 Field Documentation . . . . .	382
8.388.2.1 spc . . . . .	382
8.389pack_dms_SetActivationStatusCallback_t Struct Reference . . . . .	382
8.389.1 Detailed Description . . . . .	382
8.389.2 Field Documentation . . . . .	382
8.389.2.1 activationState . . . . .	382
8.390pack_dms_SetCrashAction_t Struct Reference . . . . .	382
8.390.1 Detailed Description . . . . .	383
8.390.2 Field Documentation . . . . .	383

8.390.2.1 crashAction . . . . .	383
8.391pack_dms_SetCustFeature_t Struct Reference . . . . .	383
8.391.1 Detailed Description . . . . .	383
8.391.2 Field Documentation . . . . .	385
8.391.2.1 DHCPRelayEnabled . . . . .	385
8.391.2.2 DisableIMSI . . . . .	385
8.391.2.3 GpsEnable . . . . .	385
8.391.2.4 GPSLPM . . . . .	385
8.391.2.5 GPSSel . . . . .	385
8.391.2.6 IPFamSupport . . . . .	385
8.391.2.7 IsVoiceEnabled . . . . .	385
8.391.2.8 RMAutoConnect . . . . .	385
8.391.2.9 SMSSupport . . . . .	385
8.392pack_dms_SetCustFeaturesV2_t Struct Reference . . . . .	385
8.392.1 Detailed Description . . . . .	385
8.392.2 Field Documentation . . . . .	386
8.392.2.1 cust_id . . . . .	386
8.392.2.2 cust_value . . . . .	386
8.392.2.3 Tlvresult . . . . .	386
8.392.2.4 value_length . . . . .	386
8.393pack_dms_SetEventReport_t Struct Reference . . . . .	386
8.393.1 Detailed Description . . . . .	386
8.393.2 Field Documentation . . . . .	386
8.393.2.1 mode . . . . .	386
8.394pack_dms_SetIndicationRegister_t Struct Reference . . . . .	386
8.394.1 Detailed Description . . . . .	386
8.394.2 Field Documentation . . . . .	387
8.394.2.1 PSMCfgChangeInfo . . . . .	387
8.394.2.2 PSMStatus . . . . .	387
8.394.2.3 RptIMSCapability . . . . .	387
8.395pack_dms_SetPower_t Struct Reference . . . . .	387
8.395.1 Detailed Description . . . . .	387
8.395.2 Field Documentation . . . . .	388
8.395.2.1 mode . . . . .	388
8.395.2.2 Tlvresult . . . . .	388
8.396pack_dms_SetUSBComp_t Struct Reference . . . . .	388
8.396.1 Detailed Description . . . . .	388
8.396.2 Field Documentation . . . . .	388
8.396.2.1 Tlvresult . . . . .	388
8.396.2.2 USBComp . . . . .	389



8.397	<a href="#">pack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference</a>	389
8.397.1	Detailed Description	389
8.397.2	Field Documentation	389
8.397.2.1	resetInfoInd	389
8.398	<a href="#">pack_dms_SLQSSetPowerSaveModeConfig_t Struct Reference</a>	389
8.398.1	Detailed Description	389
8.398.2	Field Documentation	390
8.398.2.1	pActiveTimer	390
8.398.2.2	pPeriodicUpdateTimer	390
8.398.2.3	pPsmEnableState	390
8.399	<a href="#">pack_dms_SLQSSwiGetCrashInfo_t Struct Reference</a>	390
8.399.1	Detailed Description	390
8.399.2	Field Documentation	390
8.399.2.1	clear	390
8.400	<a href="#">pack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference</a>	390
8.400.1	Detailed Description	390
8.400.2	Field Documentation	391
8.400.2.1	pDestSMSContent	391
8.400.2.2	pDestSMSNum	391
8.401	<a href="#">pack_dms_SLQSSwiSetHostDevInfo_t Struct Reference</a>	391
8.401.1	Detailed Description	391
8.401.2	Field Documentation	392
8.401.2.1	hostID	392
8.401.2.2	manString	392
8.401.2.3	modelString	392
8.401.2.4	plasmaIDString	392
8.401.2.5	swVerString	392
8.402	<a href="#">pack_dms_SLQSSwiSetOSInfo_t Struct Reference</a>	392
8.402.1	Detailed Description	392
8.402.2	Field Documentation	392
8.402.2.1	nameString	392
8.402.2.2	versionString	392
8.403	<a href="#">pack_dms_SwiSetEventReport_t Struct Reference</a>	392
8.403.1	Detailed Description	392
8.403.2	Field Documentation	393
8.403.2.1	pTempReport	393
8.403.2.2	pUIMStatusReport	393
8.403.2.3	pVoltReport	393
8.404	<a href="#">pack_dms_SwiUimSelect_t Struct Reference</a>	393
8.404.1	Detailed Description	393

8.404.2 Field Documentation . . . . .	393
8.404.2.1 uim_select . . . . .	394
8.405pack_dms_UIMChangePIN_t Struct Reference . . . . .	394
8.405.1 Detailed Description . . . . .	394
8.405.2 Field Documentation . . . . .	394
8.405.2.1 id . . . . .	394
8.405.2.2 newValue . . . . .	394
8.405.2.3 oldValue . . . . .	394
8.406pack_dms_UIMGetControlKeyStatus_t Struct Reference . . . . .	394
8.406.1 Detailed Description . . . . .	394
8.406.2 Field Documentation . . . . .	395
8.406.2.1 facility . . . . .	395
8.407pack_dms_UIMGetICCID_t Struct Reference . . . . .	395
8.407.1 Detailed Description . . . . .	395
8.407.2 Field Documentation . . . . .	395
8.407.2.1 ParamPresenceMask . . . . .	395
8.407.2.2 Tlvresult . . . . .	395
8.408pack_dms_UIMSetControlKeyProtection_t Struct Reference . . . . .	395
8.408.1 Detailed Description . . . . .	395
8.408.2 Field Documentation . . . . .	396
8.408.2.1 facility . . . . .	396
8.408.2.2 facilityCk . . . . .	396
8.408.2.3 facilityState . . . . .	396
8.409pack_dms_UIMSetPINProtection_t Struct Reference . . . . .	396
8.409.1 Detailed Description . . . . .	396
8.409.2 Field Documentation . . . . .	397
8.409.2.1 bEnable . . . . .	397
8.409.2.2 id . . . . .	397
8.409.2.3 value . . . . .	397
8.410pack_dms_UIMUnblockControlKey_t Struct Reference . . . . .	397
8.410.1 Detailed Description . . . . .	397
8.410.2 Field Documentation . . . . .	397
8.410.2.1 facility . . . . .	397
8.410.2.2 facilityCk . . . . .	397
8.411pack_dms_UIMUnblockPIN_t Struct Reference . . . . .	397
8.411.1 Detailed Description . . . . .	397
8.411.2 Field Documentation . . . . .	398
8.411.2.1 id . . . . .	398
8.411.2.2 newPin . . . . .	398
8.411.2.3 pukValue . . . . .	398

8.412pack_dms_UIMVerifyPIN_t Struct Reference . . . . .	398
8.412.1 Detailed Description . . . . .	398
8.412.2 Field Documentation . . . . .	398
8.412.2.1 id . . . . .	398
8.412.2.2 value . . . . .	398
8.413pack_fms_GetImagesPreference_t Struct Reference . . . . .	398
8.413.1 Detailed Description . . . . .	399
8.413.2 Field Documentation . . . . .	399
8.413.2.1 Tlvresult . . . . .	399
8.414pack_fms_GetStoredImages_t Struct Reference . . . . .	399
8.414.1 Detailed Description . . . . .	399
8.414.2 Field Documentation . . . . .	399
8.414.2.1 Tlvresult . . . . .	399
8.415pack_fms_SetImagesPreference_t Struct Reference . . . . .	399
8.415.1 Detailed Description . . . . .	399
8.415.2 Field Documentation . . . . .	400
8.415.2.1 bForceDownload . . . . .	400
8.415.2.2 imageListSize . . . . .	400
8.415.2.3 modemindex . . . . .	400
8.415.2.4 pImageList . . . . .	400
8.415.2.5 Tlvresult . . . . .	400
8.416pack_ims_SLQSImsConfigIndicationRegister_t Struct Reference . . . . .	400
8.416.1 Detailed Description . . . . .	400
8.416.2 Field Documentation . . . . .	401
8.416.2.1 pRegMgrConfigEvents . . . . .	401
8.416.2.2 pSIPConfigEvents . . . . .	401
8.416.2.3 pSMSConfigEvents . . . . .	401
8.416.2.4 pUserConfigEvents . . . . .	401
8.416.2.5 pVoIPConfigEvents . . . . .	401
8.417pack_ims_SLQSSetIMSSMSConfig_t Struct Reference . . . . .	401
8.417.1 Detailed Description . . . . .	402
8.417.2 Field Documentation . . . . .	402
8.417.2.1 pPhoneCtxtURI . . . . .	402
8.417.2.2 pPhoneCtxtURILen . . . . .	402
8.417.2.3 pSMSFormat . . . . .	402
8.417.2.4 pSMSOverIPNwInd . . . . .	402
8.418pack_ims_SLQSSetIMSUserConfig_t Struct Reference . . . . .	402
8.418.1 Detailed Description . . . . .	402
8.418.2 Field Documentation . . . . .	403
8.418.2.1 pIMSDomain . . . . .	403

8.418.2.2 pIMSDomainLen . . . . .	403
8.419pack_ims_SLQSSetIMSVoIPConfig_t Struct Reference . . . . .	403
8.419.1 Detailed Description . . . . .	403
8.419.2 Field Documentation . . . . .	405
8.419.2.1 pAmrMode . . . . .	405
8.419.2.2 pAmrOctetAligned . . . . .	405
8.419.2.3 pAmrWbEnable . . . . .	405
8.419.2.4 pAmrWBMode . . . . .	405
8.419.2.5 pAmrWBOctetAligned . . . . .	405
8.419.2.6 pMinSessionExpiryTimer . . . . .	405
8.419.2.7 pRingBackTimer . . . . .	405
8.419.2.8 pRingingTimer . . . . .	405
8.419.2.9 pRTPRTCPInactTimer . . . . .	405
8.419.2.10pScrAmrEnable . . . . .	405
8.419.2.11pScrAmrWbEnable . . . . .	405
8.419.2.12pSessionExpiryTimer . . . . .	405
8.420pack_ims_SLQSSetRegMgrConfig_t Struct Reference . . . . .	405
8.420.1 Detailed Description . . . . .	405
8.420.2 Field Documentation . . . . .	406
8.420.2.1 pCSCFPortName . . . . .	406
8.420.2.2 pCSCFPortNameLen . . . . .	406
8.420.2.3 pIMSTestMode . . . . .	406
8.420.2.4 pPriCSCFPort . . . . .	406
8.421pack_ims_SLQSSetSIPConfig_t Struct Reference . . . . .	406
8.421.1 Detailed Description . . . . .	406
8.421.2 Field Documentation . . . . .	406
8.421.2.1 pSigCompEnabled . . . . .	407
8.421.2.2 pSIPLocalPort . . . . .	407
8.421.2.3 pSubscribeTimer . . . . .	407
8.421.2.4 pTimerSIPReg . . . . .	407
8.421.2.5 pTimerT1 . . . . .	407
8.421.2.6 pTimerT2 . . . . .	407
8.421.2.7 pTimerTf . . . . .	407
8.422pack_imsa_SLQSRegisterIMSAIndication_t Struct Reference . . . . .	407
8.422.1 Detailed Description . . . . .	407
8.422.2 Field Documentation . . . . .	408
8.422.2.1 has_PdpStatusConfig . . . . .	408
8.422.2.2 has_RatHandoverStatusConfig . . . . .	408
8.422.2.3 has_RegStatusConfig . . . . .	408
8.422.2.4 has_ServiceStatusConfig . . . . .	408

8.422.2.5 PdpStatusConfig . . . . .	408
8.422.2.6 RatHandoverStatusConfig . . . . .	408
8.422.2.7 RegStatusConfig . . . . .	408
8.422.2.8 ServiceStatusConfig . . . . .	408
8.423pack_loc_Delete_Assist_Data_t Struct Reference . . . . .	408
8.423.1 Detailed Description . . . . .	408
8.423.2 Field Documentation . . . . .	409
8.423.2.1 pBdsSVInfo . . . . .	409
8.423.2.2 pCellDb . . . . .	409
8.423.2.3 pClkInfo . . . . .	409
8.423.2.4 pGnssData . . . . .	409
8.423.2.5 pSVInfo . . . . .	409
8.423.2.6 Tlvresult . . . . .	409
8.424pack_loc_EventRegister_t Struct Reference . . . . .	409
8.424.1 Detailed Description . . . . .	409
8.424.2 Field Documentation . . . . .	411
8.424.2.1 eventRegister . . . . .	411
8.424.2.2 Tlvresult . . . . .	411
8.425pack_loc_SetExtPowerState_t Struct Reference . . . . .	411
8.425.1 Detailed Description . . . . .	411
8.425.2 Field Documentation . . . . .	411
8.425.2.1 extPowerState . . . . .	411
8.425.2.2 Tlvresult . . . . .	411
8.426pack_loc_SetOperationMode_t Struct Reference . . . . .	411
8.426.1 Detailed Description . . . . .	412
8.426.2 Field Documentation . . . . .	412
8.426.2.1 mode . . . . .	412
8.426.2.2 Tlvresult . . . . .	412
8.427pack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . . .	412
8.427.1 Detailed Description . . . . .	412
8.427.2 Field Documentation . . . . .	412
8.427.2.1 Tlvresult . . . . .	412
8.427.2.2 xid . . . . .	413
8.428pack_loc_SLQSLOCGetServer_t Struct Reference . . . . .	413
8.428.1 Detailed Description . . . . .	413
8.428.2 Field Documentation . . . . .	413
8.428.2.1 has_serverAddrTypeMask . . . . .	413
8.428.2.2 serverAddrTypeMask . . . . .	413
8.428.2.3 serverType . . . . .	413
8.429pack_loc_SLQSLOCInjectPosition_t Struct Reference . . . . .	413

8.429.1 Detailed Description	414
8.429.2 Field Documentation	417
8.429.2.1 altitudeSrcInfo	417
8.429.2.2 altitudeWrtEllipsoid	417
8.429.2.3 altitudeWrtMeanSeaLevel	417
8.429.2.4 has_altitudeSrcInfo	417
8.429.2.5 has_altitudeWrtEllipsoid	417
8.429.2.6 has_altitudeWrtMeanSeaLevel	417
8.429.2.7 has_horConfidence	417
8.429.2.8 has_horReliability	417
8.429.2.9 has_horUncCircular	417
8.429.2.10 has_latitude	417
8.429.2.11 has_longitude	417
8.429.2.12 has_positionSrc	417
8.429.2.13 has_rawHorConfidence	417
8.429.2.14 has_rawHorUncCircular	417
8.429.2.15 has_timestampAge	417
8.429.2.16 has_timestampUtc	417
8.429.2.17 has_vertConfidence	417
8.429.2.18 has_vertReliability	418
8.429.2.19 has_vertUnc	418
8.429.2.20 horConfidence	418
8.429.2.21 horReliability	418
8.429.2.22 horUncCircular	418
8.429.2.23 latitude	418
8.429.2.24 longitude	418
8.429.2.25 positionSrc	418
8.429.2.26 rawHorConfidence	418
8.429.2.27 rawHorUncCircular	418
8.429.2.28 timestampAge	418
8.429.2.29 timestampUtc	418
8.429.2.30 vertConfidence	418
8.429.2.31 vertReliability	418
8.429.2.32 vertUnc	418
8.430 pack_loc_SLQSLOCInjectSensorData_t Struct Reference	418
8.430.1 Detailed Description	418
8.430.2 Field Documentation	419
8.430.2.1 acceleroData	419
8.430.2.2 acceleroTimeSrc	419
8.430.2.3 accelTemp	419

8.430.2.4 gyroData . . . . .	419
8.430.2.5 gyroTemp . . . . .	420
8.430.2.6 gyroTimeSrc . . . . .	420
8.430.2.7 has_acceleroTimeSrc . . . . .	420
8.430.2.8 has_accelTemp . . . . .	420
8.430.2.9 has_accleroData . . . . .	420
8.430.2.10has_gyroData . . . . .	420
8.430.2.11has_gyroTemp . . . . .	420
8.430.2.12has_gyroTimeSrc . . . . .	420
8.430.2.13has_opaqueId . . . . .	420
8.430.2.14opaqueId . . . . .	420
8.431pack_loc_SLQSLOCInjectUTCTime_t Struct Reference . . . . .	420
8.431.1 Detailed Description . . . . .	420
8.431.2 Field Documentation . . . . .	420
8.431.2.1 timeMsec . . . . .	420
8.431.2.2 timeUncMsec . . . . .	420
8.432pack_loc_SLQSLOCSetCradleMountConfig_t Struct Reference . . . . .	420
8.432.1 Detailed Description . . . . .	421
8.432.2 Field Documentation . . . . .	421
8.432.2.1 confidence . . . . .	421
8.432.2.2 has_confidence . . . . .	421
8.432.2.3 state . . . . .	421
8.433pack_loc_SLQSLOCSetServer_t Struct Reference . . . . .	421
8.433.1 Detailed Description . . . . .	421
8.433.2 Field Documentation . . . . .	422
8.433.2.1 pIPv4Config . . . . .	422
8.433.2.2 pIPv6Config . . . . .	422
8.433.2.3 pURLAddr . . . . .	422
8.433.2.4 serverType . . . . .	422
8.434pack_loc_Start_t Struct Reference . . . . .	422
8.434.1 Detailed Description . . . . .	422
8.434.2 Field Documentation . . . . .	423
8.434.2.1 pApplicationInfo . . . . .	423
8.434.2.2 pConfigAltitudeAssumed . . . . .	423
8.434.2.3 pHorizontalAccuracyLvl . . . . .	423
8.434.2.4 pIntermediateReportState . . . . .	423
8.434.2.5 pMinIntervalTime . . . . .	423
8.434.2.6 pRecurrenceType . . . . .	424
8.434.2.7 SessionId . . . . .	424
8.434.2.8 Tlvresult . . . . .	424

8.435	<a href="#">pack_loc_Stop_t Struct Reference</a>	424
8.435.1	Detailed Description	424
8.435.2	Field Documentation	424
8.435.2.1	SessionId	424
8.435.2.2	Tlvresult	424
8.436	<a href="#">pack_nas_InitiateDomainAttach_t Struct Reference</a>	424
8.436.1	Detailed Description	424
8.436.2	Field Documentation	424
8.436.2.1	action	425
8.437	<a href="#">pack_nas_PerformNetworkScanPCI_t Struct Reference</a>	425
8.437.1	Detailed Description	425
8.437.2	Field Documentation	427
8.437.2.1	pCiotOpModePref	427
8.437.2.2	pLteM1BandPref	428
8.437.2.3	pLteNB1BandPref	428
8.437.2.4	pNetworkType	428
8.437.2.5	pScanType	428
8.438	<a href="#">pack_nas_SetACCOLC_t Struct Reference</a>	428
8.438.1	Detailed Description	428
8.438.2	Field Documentation	428
8.438.2.1	accolc	428
8.438.2.2	spc	428
8.439	<a href="#">pack_nas_SetCDMANetworkParameters_t Struct Reference</a>	428
8.439.1	Detailed Description	428
8.439.2	Field Documentation	429
8.439.2.1	pApplication	429
8.439.2.2	pBroadcast	429
8.439.2.3	pCustomSCP	430
8.439.2.4	pForceRev0	430
8.439.2.5	pProtocol	430
8.439.2.6	pRoaming	430
8.439.2.7	pSPC	430
8.440	<a href="#">pack_nas_SetNetworkPreference_t Struct Reference</a>	430
8.440.1	Detailed Description	430
8.440.2	Field Documentation	430
8.440.2.1	Duration	430
8.440.2.2	TechnologyPref	431
8.440.2.3	Tlvresult	431
8.441	<a href="#">pack_nas_SLQSCfgSigInfo_t Struct Reference</a>	431
8.441.1	Detailed Description	431



8.441.2 Field Documentation . . . . .	432
8.441.2.1 pECIOThresh . . . . .	432
8.441.2.2 pHDRSINRThresh . . . . .	432
8.441.2.3 pIOThresh . . . . .	432
8.441.2.4 pLTESigRptCfg . . . . .	432
8.441.2.5 pLTESNRThresh . . . . .	432
8.441.2.6 pRSRPThresh . . . . .	432
8.441.2.7 pRSRQThresh . . . . .	432
8.441.2.8 pRSSIThresh . . . . .	432
8.441.2.9 pTDSCDMASINRCONFThresh . . . . .	432
8.442 pack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	432
8.442.1 Detailed Description . . . . .	432
8.442.2 Field Documentation . . . . .	433
8.442.2.1 mcc . . . . .	433
8.442.2.2 mnc . . . . .	433
8.442.2.3 pMncPcsStatus . . . . .	433
8.443 pack_nas_SLQSInitiateNetworkRegistration_t Struct Reference . . . . .	433
8.443.1 Detailed Description . . . . .	433
8.443.2 Field Documentation . . . . .	434
8.443.2.1 pChangeDuration . . . . .	434
8.443.2.2 pMncPcsDigitStatus . . . . .	434
8.443.2.3 pMNRInfo . . . . .	434
8.443.2.4 regAction . . . . .	434
8.444 pack_nas_SLQSNasConfigSigInfo2_t Struct Reference . . . . .	434
8.444.1 Detailed Description . . . . .	435
8.444.2 Field Documentation . . . . .	437
8.444.2.1 pCDMAECIODelta . . . . .	437
8.444.2.2 pCDMAECIOThresh . . . . .	437
8.444.2.3 pCDMARSSIDelta . . . . .	437
8.444.2.4 pCDMARSSIThresh . . . . .	437
8.444.2.5 pGSMRSSIDelta . . . . .	437
8.444.2.6 pGSMRSSIThresh . . . . .	437
8.444.2.7 pHDRECIODelta . . . . .	437
8.444.2.8 pHDRECIOTThresh . . . . .	438
8.444.2.9 pHDRIODelta . . . . .	438
8.444.2.10 pHDRIOThresh . . . . .	438
8.444.2.11 pHDRRSSIDelta . . . . .	438
8.444.2.12 pHDRRSSIThresh . . . . .	438
8.444.2.13 pHDRSINRDelta . . . . .	438
8.444.2.14 pHDRSINRThresh . . . . .	438

8.444.2.15	<a href="#">pLTERSRPDelta</a>	438
8.444.2.16	<a href="#">pLTERSRPThresh</a>	438
8.444.2.17	<a href="#">pLTERSRQDelta</a>	438
8.444.2.18	<a href="#">pLTERSRQThresh</a>	438
8.444.2.19	<a href="#">pLTERSSIDelta</a>	438
8.444.2.20	<a href="#">pLTERSSIThresh</a>	438
8.444.2.21	<a href="#">pLTESigRptConfig</a>	438
8.444.2.22	<a href="#">pLTESNRDelta</a>	438
8.444.2.23	<a href="#">pLTESNRThresh</a>	438
8.444.2.24	<a href="#">pTDSCDMAECIODelta</a>	438
8.444.2.25	<a href="#">pTDSCDMAECIOThresh</a>	438
8.444.2.26	<a href="#">pTDSCDMARSCPDelta</a>	438
8.444.2.27	<a href="#">pTDSCDMARSCPThresh</a>	438
8.444.2.28	<a href="#">pTDSCDMARSSIDelta</a>	438
8.444.2.29	<a href="#">pTDSCDMARSSIThresh</a>	438
8.444.2.30	<a href="#">pTDSCDMASINRDelta</a>	438
8.444.2.31	<a href="#">pTDSCDMASINRThresh</a>	438
8.444.2.32	<a href="#">pWCDMAECIODelta</a>	438
8.444.2.33	<a href="#">pWCDMAECIOThresh</a>	438
8.444.2.34	<a href="#">pWCDMARSSIDelta</a>	438
8.444.2.35	<a href="#">pWCDMARSSIThresh</a>	438
8.445	<a href="#">pack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference</a>	439
8.445.1	Detailed Description	439
8.445.2	Field Documentation	439
8.445.2.1	namID	439
8.446	<a href="#">pack_nas_SLQSNASGeteDRXParamsExt_t Struct Reference</a>	439
8.446.1	Detailed Description	439
8.446.2	Field Documentation	440
8.446.2.1	pEdrxRAT	440
8.446.2.2	pLteOpMode	440
8.447	<a href="#">pack_nas_SLQSNasGetTxRxInfo_t Struct Reference</a>	440
8.447.1	Detailed Description	440
8.447.2	Field Documentation	440
8.447.2.1	radio_if	440
8.448	<a href="#">pack_nas_SLQSNasIndicationRegisterExt_t Struct Reference</a>	440
8.448.1	Detailed Description	441
8.448.2	Field Documentation	443
8.448.2.1	pDDTMInd	443
8.448.2.2	pDualStandByPrefInd	443
8.448.2.3	pEdrxChangeInfoInd	443

8.448.2.4 pErrorRateInd . . . . .	443
8.448.2.5 pHDRNewUATIAssInd . . . . .	443
8.448.2.6 pHDRSessionCloseInd . . . . .	443
8.448.2.7 pLTECphyCa . . . . .	443
8.448.2.8 pManagedRoamingInd . . . . .	443
8.448.2.9 pNetworkRejectInd . . . . .	443
8.448.2.10pNetworkTimeInd . . . . .	443
8.448.2.11pServingSystemInd . . . . .	443
8.448.2.12pSignalStrengthInd . . . . .	443
8.448.2.13pSubscriptionInfoInd . . . . .	443
8.448.2.14pSuppressSysInfoInd . . . . .	443
8.448.2.15pSysInfoInd . . . . .	443
8.448.2.16pSystemSelectionInd . . . . .	443
8.449pack_nas_SLQSNasIndicationRegisterV2_t Struct Reference . . . . .	443
8.449.1 Detailed Description . . . . .	444
8.449.2 Field Documentation . . . . .	450
8.449.2.1 pAccessClassBarringInd . . . . .	450
8.449.2.2 pCallModeStatusInd . . . . .	450
8.449.2.3 pCSPPLMNModelInd . . . . .	450
8.449.2.4 pCurrentPLMNNameInd . . . . .	450
8.449.2.5 pDataSubscriptionPriorityInd . . . . .	450
8.449.2.6 pDDTMInd . . . . .	450
8.449.2.7 pDualStandByPrefInd . . . . .	450
8.449.2.8 pE911StateReadyStatusInd . . . . .	450
8.449.2.9 pEdrxChangeInfoInd . . . . .	450
8.449.2.10pEMBMSStatusInd . . . . .	450
8.449.2.11pEmergencyModeStatusInd . . . . .	450
8.449.2.12pErrorRateInd . . . . .	450
8.449.2.13pGcellInfoInd . . . . .	450
8.449.2.14pHDRNewUATIAssInd . . . . .	450
8.449.2.15pHDRSessionCloseInd . . . . .	450
8.449.2.16pIMSPrefStatusInd . . . . .	450
8.449.2.17pLTECphyCa . . . . .	451
8.449.2.18pLTERACHFailInd . . . . .	451
8.449.2.19pLTERRCTxInfoInd . . . . .	451
8.449.2.20pLTESIB16NetworkTimeInd . . . . .	451
8.449.2.21pManagedRoamingInd . . . . .	451
8.449.2.22pManualNWScanFailureInd . . . . .	451
8.449.2.23pNetworkRejectInd . . . . .	451
8.449.2.24pNetworkTimeInd . . . . .	451

8.449.2.25	<a href="#">pOperatorNameDataInd</a>	451
8.449.2.26	<a href="#">pRFBandInfoInd</a>	451
8.449.2.27	<a href="#">pRTREConfigurationInd</a>	451
8.449.2.28	<a href="#">pServingSystemInd</a>	451
8.449.2.29	<a href="#">pSignalStrengthInd</a>	451
8.449.2.30	<a href="#">pSSACBarringExtInd</a>	451
8.449.2.31	<a href="#">pSSACBarringInd</a>	451
8.449.2.32	<a href="#">pSubscriptionChangeInd</a>	451
8.449.2.33	<a href="#">pSubscriptionInfoInd</a>	451
8.449.2.34	<a href="#">pSuppressSysInfoInd</a>	451
8.449.2.35	<a href="#">pSysInfoInd</a>	451
8.449.2.36	<a href="#">pSystemSelectionInd</a>	451
8.449.2.37	<a href="#">pT3346TimerStatusChangeInd</a>	451
8.449.2.38	<a href="#">pT3402TimerValueInd</a>	451
8.449.2.39	<a href="#">pTimerExpiryInd</a>	451
8.450	<a href="#">pack_nas_SLQSNASSeteDRXParams_t Struct Reference</a>	451
8.450.1	<a href="#">Detailed Description</a>	452
8.450.2	<a href="#">Field Documentation</a>	452
8.450.2.1	<a href="#">pCycleLen</a>	452
8.450.2.2	<a href="#">pEdrxCiotLteMode</a>	453
8.450.2.3	<a href="#">pEdrxEnable</a>	453
8.450.2.4	<a href="#">pEdrxRatType</a>	453
8.450.2.5	<a href="#">pPagingTimeWindow</a>	453
8.451	<a href="#">pack_nas_SLQSNasSwiIndicationRegister_t Struct Reference</a>	453
8.451.1	<a href="#">Detailed Description</a>	453
8.451.2	<a href="#">Field Documentation</a>	454
8.451.2.1	<a href="#">gsmUmtsDI</a>	454
8.451.2.2	<a href="#">gsmUmtsUI</a>	454
8.451.2.3	<a href="#">lteEmmDI</a>	454
8.451.2.4	<a href="#">lteEmmUI</a>	454
8.451.2.5	<a href="#">lteEsmDI</a>	454
8.451.2.6	<a href="#">lteEsmUI</a>	454
8.451.2.7	<a href="#">pRankIndicatorInd</a>	454
8.451.2.8	<a href="#">pTimer</a>	454
8.452	<a href="#">pack_nas_SLQSNASSwiSetChannelLock_t Struct Reference</a>	454
8.452.1	<a href="#">Detailed Description</a>	454
8.452.2	<a href="#">Field Documentation</a>	454
8.452.2.1	<a href="#">pLteEARFCN</a>	454
8.452.2.2	<a href="#">pLtePCI</a>	455
8.452.2.3	<a href="#">pWcdmaUARFCN</a>	455

8.453	<a href="#">pack_nas_SLQSSetBandPreference_t Struct Reference</a>	455
8.453.1	Detailed Description	455
8.453.2	Field Documentation	456
8.453.2.1	bandPref	457
8.454	<a href="#">pack_nas_SLQSSetSignalStrengthsCallback_t Struct Reference</a>	457
8.454.1	Detailed Description	457
8.454.2	Field Documentation	457
8.454.2.1	bEnable	457
8.454.2.2	pSigIndReq	457
8.455	<a href="#">pack_nas_SLQSSetSysSelectionPref_t Struct Reference</a>	457
8.455.1	Detailed Description	457
8.455.2	Field Documentation	462
8.455.2.1	pAcqOrderPref	462
8.455.2.2	pBandPref	462
8.455.2.3	pChgDuration	462
8.455.2.4	pCSGID	462
8.455.2.5	pEmerMode	462
8.455.2.6	pGWAcqOrderPref	462
8.455.2.7	pLTEBandPref	462
8.455.2.8	pMNCIncPCSDigStat	462
8.455.2.9	pModePref	462
8.455.2.10	pNetSelPref	462
8.455.2.11	pPRLPref	462
8.455.2.12	pRAT	462
8.455.2.13	pRoamPref	462
8.455.2.14	pSrvDomainPref	462
8.455.2.15	pSrvRegRestriction	463
8.455.2.16	pTdsdmaBandPref	463
8.456	<a href="#">pack_nas_SLQSSetSysSelectionPrefExt_t Struct Reference</a>	463
8.456.1	Detailed Description	463
8.456.2	Field Documentation	472
8.456.2.1	pAcqOrderPref	472
8.456.2.2	pBandPref	472
8.456.2.3	pChgDuration	472
8.456.2.4	pCiotAcqOrderPref	472
8.456.2.5	pCiotLteOpMode	472
8.456.2.6	pCiotLteOpModePref	472
8.456.2.7	pCSGID	472
8.456.2.8	pEmerMode	472
8.456.2.9	pGWAcqOrderPref	472

8.456.2.10pLTEBandPref . . . . .	472
8.456.2.11pLTEBandPrefExt . . . . .	472
8.456.2.12pLteM1BandPref . . . . .	472
8.456.2.13pLteNb1BandPref . . . . .	472
8.456.2.14pMNCIncPCSDigStat . . . . .	472
8.456.2.15pModePref . . . . .	472
8.456.2.16pNetSelPref . . . . .	472
8.456.2.17pNr5gBandPref . . . . .	472
8.456.2.18pPRLPref . . . . .	473
8.456.2.19pRAT . . . . .	473
8.456.2.20pRoamPref . . . . .	473
8.456.2.21pSrvDomainPref . . . . .	473
8.456.2.22pSrvRegRestriction . . . . .	473
8.456.2.23pTdsdmaBandPref . . . . .	473
8.457pack_nas_SLQSSwiPSDetach_t Struct Reference . . . . .	473
8.457.1 Detailed Description . . . . .	473
8.457.2 Field Documentation . . . . .	473
8.457.2.1 pDetachAction . . . . .	473
8.458pack_pds_PDSInjectTimeReference_t Struct Reference . . . . .	473
8.458.1 Detailed Description . . . . .	473
8.458.2 Field Documentation . . . . .	474
8.458.2.1 systemDiscontinuities . . . . .	474
8.458.2.2 systemTime . . . . .	474
8.459pack_pds_ResetPDSData_t Struct Reference . . . . .	474
8.459.1 Detailed Description . . . . .	474
8.459.2 Field Documentation . . . . .	474
8.459.2.1 pCellDataMask . . . . .	475
8.459.2.2 pGPSDataMask . . . . .	475
8.460pack_pds_SetEventReportCallback_t Struct Reference . . . . .	475
8.460.1 Detailed Description . . . . .	475
8.460.2 Field Documentation . . . . .	475
8.460.2.1 posDataNmea . . . . .	475
8.460.2.2 rptPosData . . . . .	475
8.461pack_pds_SetPDSDefaults_t Struct Reference . . . . .	475
8.461.1 Detailed Description . . . . .	475
8.461.2 Field Documentation . . . . .	476
8.461.2.1 accuracy . . . . .	476
8.461.2.2 interval . . . . .	476
8.461.2.3 operation . . . . .	476
8.461.2.4 timeout . . . . .	476

8.462	<a href="#">pack_pds_SetPDSSState_t Struct Reference</a>	476
8.462.1	Detailed Description	476
8.462.2	Field Documentation	476
8.462.2.1	enable	476
8.463	<a href="#">pack_pds_SetPortAutomaticTracking_t Struct Reference</a>	476
8.463.1	Detailed Description	477
8.463.2	Field Documentation	477
8.463.2.1	bAuto	477
8.464	<a href="#">pack_pds_SetServiceAutomaticTracking_t Struct Reference</a>	477
8.464.1	Detailed Description	477
8.464.2	Field Documentation	477
8.464.2.1	bAuto	477
8.465	<a href="#">pack_pds_SetXTRAAutomaticDownload_t Struct Reference</a>	477
8.465.1	Detailed Description	478
8.465.2	Field Documentation	478
8.465.2.1	bEnabled	478
8.465.2.2	interval	478
8.466	<a href="#">pack_pds_SetXTRANetwork_t Struct Reference</a>	478
8.466.1	Detailed Description	478
8.466.2	Field Documentation	478
8.466.2.1	preference	478
8.467	<a href="#">pack_pds_SLQSGetAGPSConfig_t Struct Reference</a>	478
8.467.1	Detailed Description	479
8.467.2	Field Documentation	479
8.467.2.1	pNetworkMode	479
8.468	<a href="#">pack_pds_SLQSPDSInjectAbsoluteTimeReference_t Struct Reference</a>	479
8.468.1	Detailed Description	479
8.468.2	Field Documentation	479
8.468.2.1	forceFlag	479
8.468.2.2	timeBase	479
8.468.2.3	timeMsec	480
8.468.2.4	timeUncMsec	480
8.469	<a href="#">pack_pds_SLQSPDSInjectPositionData_t Struct Reference</a>	480
8.469.1	Detailed Description	480
8.469.2	Field Documentation	481
8.469.2.1	pAltitudeWrtEllipsoid	481
8.469.2.2	pAltitudeWrtSealevel	481
8.469.2.3	pHorizontalConfidence	481
8.469.2.4	pHorizontalUncCircular	481
8.469.2.5	pLatitude	481

8.469.2.6 pLongitude . . . . .	481
8.469.2.7 pPositionSource . . . . .	481
8.469.2.8 pTimeStamp . . . . .	481
8.469.2.9 pTimeType . . . . .	481
8.469.2.10 pVerticalConfidence . . . . .	481
8.469.2.11 pVerticalUnc . . . . .	482
8.470 pack_pds_SLQSSetAGPSConfig_t Struct Reference . . . . .	482
8.470.1 Detailed Description . . . . .	482
8.470.2 Field Documentation . . . . .	482
8.470.2.1 pNetworkMode . . . . .	482
8.470.2.2 pServerAddress . . . . .	482
8.470.2.3 pServerPort . . . . .	482
8.470.2.4 pServerURL . . . . .	482
8.470.2.5 pServerURLLength . . . . .	482
8.471 pack_pds_SLQSSetPositionMethodState_t Struct Reference . . . . .	482
8.471.1 Detailed Description . . . . .	483
8.471.2 Field Documentation . . . . .	483
8.471.2.1 pWifiState . . . . .	483
8.471.2.2 pXtraDataState . . . . .	483
8.471.2.3 pXtraTimeState . . . . .	483
8.472 pack_pds_StartPDSTrackingSessionExt_t Struct Reference . . . . .	483
8.472.1 Detailed Description . . . . .	483
8.472.2 Field Documentation . . . . .	484
8.472.2.1 fixAccuracy . . . . .	484
8.472.2.2 fixCount . . . . .	484
8.472.2.3 fixInterval . . . . .	484
8.472.2.4 fixTimeout . . . . .	484
8.472.2.5 sessionControl . . . . .	484
8.472.2.6 sessionOperation . . . . .	484
8.472.2.7 sessionServerOption . . . . .	484
8.472.2.8 sessionType . . . . .	484
8.473 pack_qmi_t Struct Reference . . . . .	484
8.473.1 Detailed Description . . . . .	485
8.473.2 Field Documentation . . . . .	485
8.473.2.1 msgid . . . . .	485
8.473.2.2 svc . . . . .	485
8.473.2.3 timeout . . . . .	485
8.473.2.4 xid . . . . .	485
8.474 pack_qos_BindDataPort_t Struct Reference . . . . .	485
8.474.1 Detailed Description . . . . .	485



8.474.2 Field Documentation . . . . .	486
8.474.2.1 pMuxID . . . . .	486
8.474.2.2 pPeripheralEndPointID . . . . .	486
8.474.2.3 pSIODataPort . . . . .	486
8.475pack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . . .	486
8.475.1 Detailed Description . . . . .	486
8.475.2 Field Documentation . . . . .	486
8.475.2.1 apnId . . . . .	486
8.476pack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . . .	486
8.476.1 Detailed Description . . . . .	486
8.476.2 Field Documentation . . . . .	486
8.476.2.1 apnId . . . . .	486
8.477pack_qos_SLQSSetQosEventCallback_t Struct Reference . . . . .	486
8.477.1 Detailed Description . . . . .	487
8.477.2 Field Documentation . . . . .	487
8.477.2.1 enable . . . . .	487
8.478pack_rms_SetSMSWake_t Struct Reference . . . . .	487
8.478.1 Detailed Description . . . . .	487
8.478.2 Field Documentation . . . . .	487
8.478.2.1 enabled . . . . .	487
8.478.2.2 wake_mask . . . . .	487
8.479pack_sar_SLQSSetRfSarState_t Struct Reference . . . . .	487
8.479.1 Detailed Description . . . . .	488
8.479.2 Field Documentation . . . . .	488
8.479.2.1 RfSarState . . . . .	488
8.480pack_sms_SaveSMS_t Struct Reference . . . . .	488
8.480.1 Detailed Description . . . . .	488
8.480.2 Field Documentation . . . . .	489
8.480.2.1 messageFormat . . . . .	489
8.480.2.2 messageSize . . . . .	489
8.480.2.3 pMessage . . . . .	489
8.480.2.4 storageType . . . . .	489
8.481pack_sms_SendSMS_t Struct Reference . . . . .	489
8.481.1 Detailed Description . . . . .	489
8.481.2 Field Documentation . . . . .	489
8.481.2.1 messageFormat . . . . .	489
8.481.2.2 messageSize . . . . .	489
8.481.2.3 pLinktimer . . . . .	490
8.481.2.4 pMessage . . . . .	490
8.482pack_sms_SetNewSMSCallback_t Struct Reference . . . . .	490

8.482.1 Detailed Description	490
8.482.2 Field Documentation	490
8.482.2.1 status	490
8.483pack_sms_SetSMSCAddress_t Struct Reference	490
8.483.1 Detailed Description	490
8.483.2 Field Documentation	490
8.483.2.1 pSMSCAddress	490
8.483.2.2 pSMSCType	490
8.484pack_sms_SLQSDeleteSMS_t Struct Reference	490
8.484.1 Detailed Description	491
8.484.2 Field Documentation	491
8.484.2.1 pMessageIndex	491
8.484.2.2 pMessageMode	491
8.484.2.3 pMessageTag	491
8.484.2.4 storageType	491
8.485pack_sms_SLQSGetSMS_t Struct Reference	491
8.485.1 Detailed Description	492
8.485.2 Field Documentation	492
8.485.2.1 messageIndex	492
8.485.2.2 pMessageMode	492
8.485.2.3 storageType	492
8.486pack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference	492
8.486.1 Detailed Description	492
8.486.2 Field Documentation	492
8.486.2.1 mode	492
8.487pack_sms_SLQSGetSMSList_t Struct Reference	492
8.487.1 Detailed Description	493
8.487.2 Field Documentation	493
8.487.2.1 pMessageMode	493
8.487.2.2 pRequestedTag	493
8.487.2.3 storageType	493
8.488pack_sms_SLQSModifySMSStatus_t Struct Reference	493
8.488.1 Detailed Description	493
8.488.2 Field Documentation	494
8.488.2.1 messageIndex	494
8.488.2.2 messageTag	494
8.488.2.3 pMessageMode	494
8.488.2.4 storageType	494
8.489pack_sms_SLQSSendAsyncSMS_t Struct Reference	494
8.489.1 Detailed Description	494

8.489.2 Field Documentation . . . . .	494
8.489.2.1 pSendSmsParams . . . . .	494
8.490pack_sms_SLQSSetIndicationRegister_t Struct Reference . . . . .	494
8.490.1 Detailed Description . . . . .	495
8.490.2 Field Documentation . . . . .	495
8.490.2.1 pSetIndicationRegReq . . . . .	495
8.491pack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . . .	495
8.491.1 Detailed Description . . . . .	495
8.491.2 Field Documentation . . . . .	495
8.491.2.1 broadcastActivate . . . . .	495
8.491.2.2 mode . . . . .	495
8.492pack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference . . . . .	495
8.492.1 Detailed Description . . . . .	496
8.492.2 Field Documentation . . . . .	496
8.492.2.1 mode . . . . .	496
8.492.2.2 pBroadcastConfig . . . . .	496
8.492.2.3 pCDMABroadcastConfig . . . . .	496
8.493pack_sms_SLQSSetSmsStorage_t Struct Reference . . . . .	496
8.493.1 Detailed Description . . . . .	496
8.493.2 Field Documentation . . . . .	496
8.493.2.1 smsStorage . . . . .	497
8.494pack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference . . . . .	497
8.494.1 Detailed Description . . . . .	497
8.494.2 Field Documentation . . . . .	497
8.494.2.1 pMaxStorageSizeReq . . . . .	497
8.495pack_sms_SLQSSmsSetRoutes_t Struct Reference . . . . .	497
8.495.1 Detailed Description . . . . .	497
8.495.2 Field Documentation . . . . .	497
8.495.2.1 pSetRoutesReq . . . . .	497
8.496pack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference . . . . .	497
8.496.1 Detailed Description . . . . .	498
8.496.2 Field Documentation . . . . .	498
8.496.2.1 pGenerator . . . . .	498
8.497pack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference . . . . .	498
8.497.1 Detailed Description . . . . .	498
8.497.2 Field Documentation . . . . .	498
8.497.2.1 Generator . . . . .	498
8.497.2.2 Profile . . . . .	498
8.498pack_swiaudio_SLQSGetM2MAVMute_t Struct Reference . . . . .	498
8.498.1 Detailed Description . . . . .	498

8.498.2 Field Documentation . . . . .	499
8.498.2.1 Profile . . . . .	499
8.499pack_swiaudio_SLQSSetM2MSpkrGain_t Struct Reference . . . . .	499
8.499.1 Detailed Description . . . . .	499
8.499.2 Field Documentation . . . . .	499
8.499.2.1 Profile . . . . .	499
8.500pack_swiaudio_SLQSSetM2MAudioAVCFG_t Struct Reference . . . . .	499
8.500.1 Detailed Description . . . . .	499
8.500.2 Field Documentation . . . . .	500
8.500.2.1 Device . . . . .	500
8.500.2.2 PIFACEId . . . . .	500
8.500.2.3 pPCMPParams . . . . .	500
8.500.2.4 Profile . . . . .	500
8.501pack_swiaudio_SLQSSetM2MAudioLPBK_t Struct Reference . . . . .	500
8.501.1 Detailed Description . . . . .	500
8.501.2 Field Documentation . . . . .	500
8.501.2.1 Enable . . . . .	500
8.502pack_swiaudio_SLQSSetM2MAudioProfile_t Struct Reference . . . . .	500
8.502.1 Detailed Description . . . . .	501
8.502.2 Field Documentation . . . . .	501
8.502.2.1 pCwtMute . . . . .	501
8.502.2.2 pEarMute . . . . .	501
8.502.2.3 pGenerator . . . . .	501
8.502.2.4 pMicMute . . . . .	501
8.502.2.5 Profile . . . . .	501
8.502.2.6 pVolume . . . . .	501
8.503pack_swiaudio_SLQSSetM2MAudioVolume_t Struct Reference . . . . .	502
8.503.1 Detailed Description . . . . .	502
8.503.2 Field Documentation . . . . .	502
8.503.2.1 Generator . . . . .	502
8.503.2.2 Level . . . . .	502
8.503.2.3 Profile . . . . .	502
8.504pack_swiaudio_SLQSSetM2MAVMute_t Struct Reference . . . . .	502
8.504.1 Detailed Description . . . . .	502
8.504.2 Field Documentation . . . . .	503
8.504.2.1 EarMute . . . . .	503
8.504.2.2 MicMute . . . . .	503
8.504.2.3 pCwtMute . . . . .	503
8.504.2.4 Profile . . . . .	503
8.505pack_swiaudio_SLQSSetM2MSpkrGain_t Struct Reference . . . . .	503

8.505.1 Detailed Description . . . . .	503
8.505.2 Field Documentation . . . . .	503
8.505.2.1 Profile . . . . .	503
8.505.2.2 Value . . . . .	503
8.506pack_swiaavms_SLQSAVMSSendSelection_t Struct Reference . . . . .	504
8.506.1 Detailed Description . . . . .	504
8.506.2 Field Documentation . . . . .	504
8.506.2.1 pClientPerformOperationFlag . . . . .	504
8.506.2.2 pDeferTime . . . . .	504
8.506.2.3 pPackageID . . . . .	504
8.506.2.4 pRejectReason . . . . .	504
8.506.2.5 selection . . . . .	504
8.507pack_swiaavms_SLQSAVMSSetSettings_t Struct Reference . . . . .	504
8.507.1 Detailed Description . . . . .	505
8.507.2 Field Documentation . . . . .	506
8.507.2.1 AutoConnect . . . . .	506
8.507.2.2 AutoReboot . . . . .	506
8.507.2.3 pAPNInfo . . . . .	506
8.507.2.4 pConnectionRetryTimers . . . . .	506
8.507.2.5 pFwAutoSDM . . . . .	506
8.507.2.6 pNotifStore . . . . .	506
8.507.2.7 pPeriodInfo . . . . .	506
8.507.2.8 pPollingTimer . . . . .	506
8.507.2.9 PromptFwDownload . . . . .	506
8.507.2.10PromptFwUpdate . . . . .	506
8.508pack_swiaavms_SLQSAVMSSetSettings_v2_t Struct Reference . . . . .	506
8.508.1 Detailed Description . . . . .	506
8.508.2 Field Documentation . . . . .	507
8.508.2.1 AutoConnect . . . . .	508
8.508.2.2 pAPNInfo . . . . .	508
8.508.2.3 pAutoReboot . . . . .	508
8.508.2.4 pConnectionRetryTimers . . . . .	508
8.508.2.5 pFwAutoSDM . . . . .	508
8.508.2.6 pNotifStore . . . . .	508
8.508.2.7 pPeriodInfo . . . . .	508
8.508.2.8 pPollingTimer . . . . .	508
8.508.2.9 PromptFwDownload . . . . .	508
8.508.2.10PromptFwUpdate . . . . .	508
8.509pack_swiaavms_SLQSAVMSStartSession_t Struct Reference . . . . .	508
8.509.1 Detailed Description . . . . .	508

8.509.2 Field Documentation . . . . .	508
8.509.2.1 sessionType . . . . .	508
8.510pack_swiaoms_SLQSAVMSStopSession_t Struct Reference . . . . .	508
8.510.1 Detailed Description . . . . .	508
8.510.2 Field Documentation . . . . .	509
8.510.2.1 sessionType . . . . .	509
8.511pack_swidms_SLQSSwiDmsSetHWWatchdog_t Struct Reference . . . . .	509
8.511.1 Detailed Description . . . . .	509
8.511.2 Field Documentation . . . . .	509
8.511.2.1 enable . . . . .	509
8.511.2.2 resetDelay . . . . .	509
8.511.2.3 timeout . . . . .	509
8.512pack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . . .	509
8.512.1 Detailed Description . . . . .	510
8.512.2 Field Documentation . . . . .	510
8.512.2.1 MTUSize . . . . .	510
8.513pack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference . . . . .	510
8.513.1 Detailed Description . . . . .	510
8.513.2 Field Documentation . . . . .	511
8.513.2.1 CfgValue . . . . .	511
8.514pack_swidms_SLQSSwiDmsSetUsbNetNum_t Struct Reference . . . . .	511
8.514.1 Detailed Description . . . . .	511
8.514.2 Field Documentation . . . . .	511
8.514.2.1 nUsbNetNum . . . . .	512
8.515pack_swiloc_SwiLocSetAutoStart_t Struct Reference . . . . .	512
8.515.1 Detailed Description . . . . .	512
8.515.2 Field Documentation . . . . .	513
8.515.2.1 fix_rate . . . . .	513
8.515.2.2 fix_type . . . . .	513
8.515.2.3 function . . . . .	513
8.515.2.4 max_dist . . . . .	513
8.515.2.5 max_time . . . . .	513
8.515.2.6 set_fix_rate . . . . .	513
8.515.2.7 set_fix_type . . . . .	513
8.515.2.8 set_function . . . . .	513
8.515.2.9 set_max_dist . . . . .	513
8.515.2.10set_max_time . . . . .	513
8.516pack_swioama_SLQSOMADMCancelSession_t Struct Reference . . . . .	513
8.516.1 Detailed Description . . . . .	513
8.516.2 Field Documentation . . . . .	514

8.516.2.1 sessionType . . . . .	514
8.517pack_swioama_SLQSOMADMCancelSessionExt_t Struct Reference . . . . .	514
8.517.1 Detailed Description . . . . .	514
8.517.2 Field Documentation . . . . .	514
8.517.2.1 sessionType . . . . .	514
8.518pack_swioama_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	514
8.518.1 Detailed Description . . . . .	514
8.518.2 Field Documentation . . . . .	515
8.518.2.1 SessionType . . . . .	515
8.519pack_swioama_SLQSOMADMSendSelection_t Struct Reference . . . . .	515
8.519.1 Detailed Description . . . . .	515
8.519.2 Field Documentation . . . . .	515
8.519.2.1 pDeferTime . . . . .	515
8.519.2.2 pRejectReason . . . . .	515
8.519.2.3 selection . . . . .	515
8.520pack_swioama_SLQSOMADMSendSelectionExt_t Struct Reference . . . . .	515
8.520.1 Detailed Description . . . . .	515
8.520.2 Field Documentation . . . . .	516
8.520.2.1 selection . . . . .	516
8.521pack_swioama_SLQSOMADMSetSettings_t Struct Reference . . . . .	516
8.521.1 Detailed Description . . . . .	516
8.521.2 Field Documentation . . . . .	517
8.521.2.1 FOTAdownload . . . . .	517
8.521.2.2 FOTAUpdate . . . . .	517
8.521.2.3 pAutosdm . . . . .	517
8.521.2.4 pFwAutoCheck . . . . .	517
8.522pack_swioama_SLQSOMADMSetSettingsExt_t Struct Reference . . . . .	517
8.522.1 Detailed Description . . . . .	517
8.522.2 Field Documentation . . . . .	518
8.522.2.1 FOTAdownload . . . . .	518
8.522.2.2 FOTAUpdate . . . . .	518
8.522.2.3 FUMOEnable . . . . .	518
8.522.2.4 OMADMEEnable . . . . .	518
8.522.2.5 OMADMLogEnable . . . . .	518
8.522.2.6 PRLEnable . . . . .	518
8.523pack_swioama_SLQSOMADMStartSession_t Struct Reference . . . . .	518
8.523.1 Detailed Description . . . . .	518
8.523.2 Field Documentation . . . . .	518
8.523.2.1 sessionType . . . . .	518
8.524pack_swioama_SLQSOMADMStartSessionExt_t Struct Reference . . . . .	518

8.524.1 Detailed Description . . . . .	519
8.524.2 Field Documentation . . . . .	519
8.524.2.1 sessionType . . . . .	519
8.525pack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference . . . . .	519
8.525.1 Detailed Description . . . . .	519
8.525.2 Field Documentation . . . . .	519
8.525.2.1 mitigationDevID . . . . .	519
8.525.2.2 mitigationDevIDLen . . . . .	519
8.526pack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference . . . . .	519
8.526.1 Detailed Description . . . . .	520
8.526.2 Field Documentation . . . . .	520
8.526.2.1 mitigationDevID . . . . .	520
8.526.2.2 mitigationDevIDLen . . . . .	520
8.527pack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference . . . . .	520
8.527.1 Detailed Description . . . . .	520
8.527.2 Field Documentation . . . . .	520
8.527.2.1 mitigationDevID . . . . .	520
8.527.2.2 mitigationDevIDLen . . . . .	520
8.528pack_uim_ChangePin_t Struct Reference . . . . .	520
8.528.1 Detailed Description . . . . .	521
8.528.2 Field Documentation . . . . .	521
8.528.2.1 changePIN . . . . .	521
8.528.2.2 EncryptedPIN1 . . . . .	521
8.528.2.3 pIndicationToken . . . . .	521
8.528.2.4 pKeyReferenceID . . . . .	521
8.528.2.5 sessionInfo . . . . .	521
8.528.2.6 Tlvresult . . . . .	521
8.529pack_uim_ReadTransparent_t Struct Reference . . . . .	521
8.529.1 Detailed Description . . . . .	522
8.529.2 Field Documentation . . . . .	522
8.529.2.1 fileIndex . . . . .	522
8.529.2.2 pEncryptData . . . . .	522
8.529.2.3 pIndicationToken . . . . .	522
8.529.2.4 readTransparent . . . . .	522
8.529.2.5 sessionInfo . . . . .	522
8.529.2.6 Tlvresult . . . . .	522
8.530pack_uim_SetPinProtection_t Struct Reference . . . . .	523
8.530.1 Detailed Description . . . . .	523
8.530.2 Field Documentation . . . . .	523
8.530.2.1 EncryptedPIN1 . . . . .	523



8.530.2.2 pIndicationToken . . . . .	523
8.530.2.3 pinProtection . . . . .	523
8.530.2.4 pKeyReferenceID . . . . .	523
8.530.2.5 sessionInfo . . . . .	523
8.530.2.6 Tlvresult . . . . .	524
8.531pack_uim_SLQSUIMAuthenticate_t Struct Reference . . . . .	524
8.531.1 Detailed Description . . . . .	524
8.531.2 Field Documentation . . . . .	524
8.531.2.1 authData . . . . .	524
8.531.2.2 pIndicationToken . . . . .	524
8.531.2.3 sessionInfo . . . . .	524
8.532pack_uim_SLQSUIMDepersonalization_t Struct Reference . . . . .	524
8.532.1 Detailed Description . . . . .	524
8.532.2 Field Documentation . . . . .	525
8.532.2.1 depersonilisationInfo . . . . .	525
8.533pack_uim_SLQSUIMEventRegister_t Struct Reference . . . . .	525
8.533.1 Detailed Description . . . . .	525
8.533.2 Field Documentation . . . . .	525
8.533.2.1 eventMask . . . . .	525
8.534pack_uim_SLQSUIMGetConfiguration_t Struct Reference . . . . .	525
8.534.1 Detailed Description . . . . .	525
8.534.2 Field Documentation . . . . .	525
8.534.2.1 pConfigurationMask . . . . .	525
8.535pack_uim_SLQSUIMGetFileAttributes_t Struct Reference . . . . .	526
8.535.1 Detailed Description . . . . .	526
8.535.2 Field Documentation . . . . .	526
8.535.2.1 fileIndex . . . . .	526
8.535.2.2 pIndicationToken . . . . .	526
8.535.2.3 sessionInfo . . . . .	526
8.536pack_uim_SLQSUIMGetServiceStatus_t Struct Reference . . . . .	526
8.536.1 Detailed Description . . . . .	526
8.536.2 Field Documentation . . . . .	527
8.536.2.1 capMask . . . . .	527
8.536.2.2 sessionInfo . . . . .	527
8.537pack_uim_SLQSUIMPowerDown_t Struct Reference . . . . .	527
8.537.1 Detailed Description . . . . .	527
8.537.2 Field Documentation . . . . .	527
8.537.2.1 slot . . . . .	527
8.538pack_uim_SLQSUIMPowerUp_t Struct Reference . . . . .	527
8.538.1 Detailed Description . . . . .	527

8.538.2 Field Documentation . . . . .	528
8.538.2.1 plgnoreHotSwapSwitch . . . . .	528
8.538.2.2 slot . . . . .	528
8.539pack_uim_SLQSUIReadRecord_t Struct Reference . . . . .	528
8.539.1 Detailed Description . . . . .	528
8.539.2 Field Documentation . . . . .	529
8.539.2.1 fileIndex . . . . .	529
8.539.2.2 plndicationToken . . . . .	529
8.539.2.3 pLastRecord . . . . .	529
8.539.2.4 readRecord . . . . .	529
8.539.2.5 sessionInfo . . . . .	529
8.540pack_uim_SLQSUIRefreshComplete_t Struct Reference . . . . .	529
8.540.1 Detailed Description . . . . .	529
8.540.2 Field Documentation . . . . .	529
8.540.2.1 refreshComplete . . . . .	529
8.540.2.2 sessionInfo . . . . .	529
8.541pack_uim_SLQSUIRefreshGetLastEvent_t Struct Reference . . . . .	529
8.541.1 Detailed Description . . . . .	529
8.541.2 Field Documentation . . . . .	530
8.541.2.1 sessionInfo . . . . .	530
8.542pack_uim_SLQSUIRefreshOK_t Struct Reference . . . . .	530
8.542.1 Detailed Description . . . . .	530
8.542.2 Field Documentation . . . . .	530
8.542.2.1 OKtoRefresh . . . . .	530
8.542.2.2 sessionInfo . . . . .	530
8.543pack_uim_SLQSUIRefreshRegister_t Struct Reference . . . . .	530
8.543.1 Detailed Description . . . . .	530
8.543.2 Field Documentation . . . . .	531
8.543.2.1 regRefresh . . . . .	531
8.543.2.2 sessionInfo . . . . .	531
8.544pack_uim_SLQSUISetServiceStatus_t Struct Reference . . . . .	531
8.544.1 Detailed Description . . . . .	531
8.544.2 Field Documentation . . . . .	531
8.544.2.1 pFDNStatus . . . . .	531
8.544.2.2 sessionInfo . . . . .	531
8.545pack_uim_SLQSUISwitchSlot_t Struct Reference . . . . .	531
8.545.1 Detailed Description . . . . .	532
8.545.2 Field Documentation . . . . .	532
8.545.2.1 bLogicalSlot . . . . .	532
8.545.2.2 ulPhysicalSlot . . . . .	532

8.546	<a href="#">pack_uim_SLQSUIMWriteRecord_t Struct Reference</a>	532
8.546.1	Detailed Description	532
8.546.2	Field Documentation	533
8.546.2.1	fileIndex	533
8.546.2.2	pIndicationToken	533
8.546.2.3	sessionInfo	533
8.546.2.4	writeRecord	533
8.547	<a href="#">pack_uim_SLQSUIMWriteTransparent_t Struct Reference</a>	533
8.547.1	Detailed Description	533
8.547.2	Field Documentation	534
8.547.2.1	fileIndex	534
8.547.2.2	pIndicationToken	534
8.547.2.3	sessionInfo	534
8.547.2.4	writeTransparent	534
8.548	<a href="#">pack_uim_UnblockPin_t Struct Reference</a>	534
8.548.1	Detailed Description	534
8.548.2	Field Documentation	534
8.548.2.1	EncryptedPIN1	535
8.548.2.2	pIndicationToken	535
8.548.2.3	pinProtection	535
8.548.2.4	pKeyReferenceID	535
8.548.2.5	sessionInfo	535
8.548.2.6	Tlvresult	535
8.549	<a href="#">pack_uim_VerifyPin_t Struct Reference</a>	535
8.549.1	Detailed Description	535
8.549.2	Field Documentation	536
8.549.2.1	pEncryptedPIN1	536
8.549.2.2	pIndicationToken	536
8.549.2.3	pKeyReferenceID	536
8.549.2.4	sessionInfo	536
8.549.2.5	Tlvresult	536
8.549.2.6	verifyPIN	536
8.550	<a href="#">pack_voice_AnswerUSSD_t Struct Reference</a>	536
8.550.1	Detailed Description	536
8.550.2	Field Documentation	536
8.550.2.1	pInfo	536
8.551	<a href="#">pack_voice_OriginateUSSD_t Struct Reference</a>	536
8.551.1	Detailed Description	536
8.551.2	Field Documentation	537
8.551.2.1	pInfo	537

8.552	<a href="#">pack_voice_SLQSOriginateUSSD_t Struct Reference</a>	537
8.552.1	Detailed Description	537
8.552.2	Field Documentation	537
8.552.2.1	ussData	537
8.552.2.2	ussDCS	537
8.552.2.3	ussLen	537
8.553	<a href="#">pack_voice_SLQSVoiceALSSelectLine_t Struct Reference</a>	537
8.553.1	Detailed Description	538
8.553.2	Field Documentation	538
8.553.2.1	lineValue	538
8.554	<a href="#">pack_voice_SLQSVoiceALSSetLineSwitching_t Struct Reference</a>	538
8.554.1	Detailed Description	538
8.554.2	Field Documentation	538
8.554.2.1	switchOption	538
8.555	<a href="#">pack_voice_SLQSVoiceAnswerCall_t Struct Reference</a>	538
8.555.1	Detailed Description	538
8.555.2	Field Documentation	539
8.555.2.1	pCallId	539
8.556	<a href="#">pack_voice_SLQSVoiceBindSubscription_t Struct Reference</a>	539
8.556.1	Detailed Description	539
8.556.2	Field Documentation	539
8.556.2.1	subType	539
8.557	<a href="#">pack_voice_SLQSVoiceBurstDTMF_t Struct Reference</a>	539
8.557.1	Detailed Description	539
8.557.2	Field Documentation	539
8.557.2.1	BurstDTMFInfo	540
8.557.2.2	pBurstDTMFLengths	540
8.558	<a href="#">pack_voice_SLQSVoiceDialCall_t Struct Reference</a>	540
8.558.1	Detailed Description	540
8.558.2	Field Documentation	541
8.558.2.1	callNumber	541
8.558.2.2	pCallPartySubAdd	541
8.558.2.3	pCallType	541
8.558.2.4	pCLIRType	541
8.558.2.5	pCUGInfo	541
8.558.2.6	pEmergencyCategory	541
8.558.2.7	pSvcType	541
8.558.2.8	pUUSInfo	541
8.559	<a href="#">pack_voice_SLQSVoiceEndCall_t Struct Reference</a>	541
8.559.1	Detailed Description	542

8.559.2 Field Documentation . . . . .	542
8.559.2.1 pCallId . . . . .	542
8.560pack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . . .	542
8.560.1 Detailed Description . . . . .	542
8.560.2 Field Documentation . . . . .	542
8.560.2.1 pSvcClass . . . . .	542
8.560.2.2 reason . . . . .	543
8.561pack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . . .	543
8.561.1 Detailed Description . . . . .	543
8.561.2 Field Documentation . . . . .	543
8.561.2.1 pSvcClass . . . . .	543
8.561.2.2 Reason . . . . .	543
8.562pack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . . .	543
8.562.1 Detailed Description . . . . .	543
8.562.2 Field Documentation . . . . .	544
8.562.2.1 callID . . . . .	544
8.563pack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . . .	544
8.563.1 Detailed Description . . . . .	544
8.563.2 Field Documentation . . . . .	544
8.563.2.1 pSvcClass . . . . .	544
8.564pack_voice_SLQSVoiceGetConfig_t Struct Reference . . . . .	544
8.564.1 Detailed Description . . . . .	544
8.564.2 Field Documentation . . . . .	545
8.564.2.1 pAirTimer . . . . .	545
8.564.2.2 pAMRStatus . . . . .	546
8.564.2.3 pAutoAnswer . . . . .	546
8.564.2.4 pNamID . . . . .	546
8.564.2.5 pPrefVoicePrivacy . . . . .	546
8.564.2.6 pPrefVoiceSO . . . . .	546
8.564.2.7 pRoamTimer . . . . .	546
8.564.2.8 pTTYMode . . . . .	546
8.564.2.9 pVoiceDomainPref . . . . .	546
8.565pack_voice_SLQSVoiceIndicationRegister_t Struct Reference . . . . .	546
8.565.1 Detailed Description . . . . .	546
8.565.2 Field Documentation . . . . .	547
8.565.2.1 pRegDTMFEvents . . . . .	547
8.565.2.2 pRegVoicePrivacyEvents . . . . .	547
8.565.2.3 pSuppsNotifEvents . . . . .	547
8.566pack_voice_SLQSVoiceManageCalls_t Struct Reference . . . . .	547
8.566.1 Detailed Description . . . . .	547

8.566.2 Field Documentation . . . . .	548
8.566.2.1 pCallID . . . . .	548
8.566.2.2 SUPSType . . . . .	548
8.567pack_voice_SLQSVoiceOrigUSSDNoWait_t Struct Reference . . . . .	548
8.567.1 Detailed Description . . . . .	548
8.567.2 Field Documentation . . . . .	548
8.567.2.1 USSInformation . . . . .	548
8.568pack_voice_SLQSVoiceSendFlash_t Struct Reference . . . . .	548
8.568.1 Detailed Description . . . . .	548
8.568.2 Field Documentation . . . . .	549
8.568.2.1 pCallID . . . . .	549
8.568.2.2 pFlashPayLd . . . . .	549
8.568.2.3 pFlashType . . . . .	549
8.569pack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . . .	549
8.569.1 Detailed Description . . . . .	549
8.569.2 Field Documentation . . . . .	550
8.569.2.1 newPasswd . . . . .	550
8.569.2.2 newPasswdAgain . . . . .	550
8.569.2.3 oldPasswd . . . . .	550
8.569.2.4 Reason . . . . .	550
8.570pack_voice_SLQSVoiceSetConfig_t Struct Reference . . . . .	550
8.570.1 Detailed Description . . . . .	550
8.570.2 Field Documentation . . . . .	551
8.570.2.1 pAirTimerConfig . . . . .	551
8.570.2.2 pAutoAnswer . . . . .	551
8.570.2.3 pPrefVoiceDomain . . . . .	551
8.570.2.4 pPrefVoiceSO . . . . .	551
8.570.2.5 pRoamTimerConfig . . . . .	551
8.570.2.6 pTTYMode . . . . .	551
8.571pack_voice_SLQSVoiceSetPreferredPrivacy_t Struct Reference . . . . .	551
8.571.1 Detailed Description . . . . .	551
8.571.2 Field Documentation . . . . .	551
8.571.2.1 privacyPref . . . . .	551
8.572pack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . . .	551
8.572.1 Detailed Description . . . . .	552
8.572.2 Field Documentation . . . . .	554
8.572.2.1 pCallBarringPasswd . . . . .	554
8.572.2.2 pCallForwardingNumber . . . . .	554
8.572.2.3 pCallFwdTypeAndPlan . . . . .	554
8.572.2.4 pServiceClass . . . . .	554

8.572.2.5 pTimerVal . . . . .	554
8.572.2.6 reason . . . . .	554
8.572.2.7 voiceSvc . . . . .	554
8.573pack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . . .	554
8.573.1 Detailed Description . . . . .	554
8.573.2 Field Documentation . . . . .	554
8.573.2.1 DTMFdigit . . . . .	554
8.573.2.2 pCallID . . . . .	555
8.574pack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . . .	555
8.574.1 Detailed Description . . . . .	555
8.574.2 Field Documentation . . . . .	555
8.574.2.1 callID . . . . .	555
8.575pack_wds_DHCPv4ClientLeaseChange_t Struct Reference . . . . .	555
8.575.1 Detailed Description . . . . .	555
8.575.2 Field Documentation . . . . .	555
8.575.2.1 pEnableNotification . . . . .	555
8.576pack_wds_GetDefaultProfile_t Struct Reference . . . . .	555
8.576.1 Detailed Description . . . . .	556
8.576.2 Field Documentation . . . . .	556
8.576.2.1 profiletype . . . . .	556
8.577pack_wds_GetDefaultProfileNum_t Struct Reference . . . . .	556
8.577.1 Detailed Description . . . . .	556
8.577.2 Field Documentation . . . . .	556
8.577.2.1 family . . . . .	556
8.577.2.2 type . . . . .	556
8.578pack_wds_GetDefaultProfileV2_t Struct Reference . . . . .	556
8.578.1 Detailed Description . . . . .	557
8.578.2 Field Documentation . . . . .	557
8.578.2.1 profiletype . . . . .	557
8.579pack_wds_GetDormancyState_t Struct Reference . . . . .	557
8.579.1 Detailed Description . . . . .	557
8.580pack_wds_GetLastMobileIPError_t Struct Reference . . . . .	557
8.580.1 Detailed Description . . . . .	557
8.581pack_wds_GetMobileIP_t Struct Reference . . . . .	557
8.581.1 Detailed Description . . . . .	557
8.582pack_wds_GetMobileIPProfile_t Struct Reference . . . . .	558
8.582.1 Detailed Description . . . . .	558
8.582.2 Field Documentation . . . . .	558
8.582.2.1 index . . . . .	558
8.583pack_wds_GetPacketStatistics_t Struct Reference . . . . .	558

8.583.1 Detailed Description . . . . .	558
8.583.2 Field Documentation . . . . .	558
8.583.2.1 pStatMask . . . . .	558
8.584pack_wds_GetPacketStatus_t Struct Reference . . . . .	558
8.584.1 Detailed Description . . . . .	559
8.584.2 Field Documentation . . . . .	559
8.584.2.1 statmask . . . . .	559
8.585pack_wds_GetSessionDuration_t Struct Reference . . . . .	559
8.585.1 Detailed Description . . . . .	559
8.586pack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	559
8.586.1 Detailed Description . . . . .	559
8.586.2 Field Documentation . . . . .	560
8.586.2.1 RmTrasferStaticsReq . . . . .	560
8.587pack_wds_SetAutoconnect_t Struct Reference . . . . .	560
8.587.1 Detailed Description . . . . .	560
8.587.2 Field Documentation . . . . .	560
8.587.2.1 acroamsetting . . . . .	560
8.587.2.2 acsetting . . . . .	560
8.588pack_wds_SetDefaultProfile_t Struct Reference . . . . .	560
8.588.1 Detailed Description . . . . .	561
8.588.2 Field Documentation . . . . .	562
8.588.2.1 authentication . . . . .	562
8.588.2.2 ipAddress . . . . .	562
8.588.2.3 pApnname . . . . .	562
8.588.2.4 pdpType . . . . .	562
8.588.2.5 pName . . . . .	562
8.588.2.6 pPassword . . . . .	562
8.588.2.7 primaryDNS . . . . .	562
8.588.2.8 profileType . . . . .	562
8.588.2.9 pUsername . . . . .	562
8.588.2.10secondaryDNS . . . . .	562
8.589pack_wds_SetDefaultProfileNum_t Struct Reference . . . . .	562
8.589.1 Detailed Description . . . . .	562
8.589.2 Field Documentation . . . . .	563
8.589.2.1 family . . . . .	563
8.589.2.2 index . . . . .	563
8.589.2.3 type . . . . .	563
8.590pack_wds_SetMobileIP_t Struct Reference . . . . .	563
8.590.1 Detailed Description . . . . .	563
8.590.2 Field Documentation . . . . .	563



8.590.2.1 mode . . . . .	563
8.591pack_wds_SetMobileIPParameters_t Struct Reference . . . . .	563
8.591.1 Detailed Description . . . . .	563
8.591.2 Field Documentation . . . . .	564
8.591.2.1 pHA2002bis . . . . .	564
8.591.2.2 pHAAuthenticator . . . . .	564
8.591.2.3 pMode . . . . .	564
8.591.2.4 pReRegPeriod . . . . .	564
8.591.2.5 pReRegTraffic . . . . .	564
8.591.2.6 pRetryInterval . . . . .	564
8.591.2.7 pRetryLimit . . . . .	564
8.591.2.8 pSPC . . . . .	564
8.592pack_wds_SetMobileIPProfile_t Struct Reference . . . . .	564
8.592.1 Detailed Description . . . . .	565
8.592.2 Field Documentation . . . . .	566
8.592.2.1 index . . . . .	566
8.592.2.2 pAAASPI . . . . .	566
8.592.2.3 pAddress . . . . .	566
8.592.2.4 pEnabled . . . . .	566
8.592.2.5 pHASPI . . . . .	566
8.592.2.6 pMNAAA . . . . .	566
8.592.2.7 pMNHA . . . . .	566
8.592.2.8 pNAI . . . . .	566
8.592.2.9 pPrimaryHA . . . . .	566
8.592.2.10pRevTunneling . . . . .	566
8.592.2.11pSecondaryHA . . . . .	566
8.592.2.12spc . . . . .	566
8.593pack_wds_SLQSCreateProfile_t Struct Reference . . . . .	566
8.593.1 Detailed Description . . . . .	566
8.593.2 Field Documentation . . . . .	567
8.593.2.1 pCurProfile . . . . .	567
8.593.2.2 pProfileId . . . . .	567
8.593.2.3 pProfileType . . . . .	567
8.594pack_wds_SLQSDeleteProfile_t Struct Reference . . . . .	567
8.594.1 Detailed Description . . . . .	567
8.594.2 Field Documentation . . . . .	567
8.594.2.1 profileIndex . . . . .	568
8.594.2.2 profileType . . . . .	568
8.595pack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	568
8.595.1 Detailed Description . . . . .	568

8.596	<a href="#">pack_wds_SLQSGetDataBearerTechnology_t Struct Reference</a>	568
8.596.1	Detailed Description	568
8.597	<a href="#">pack_wds_SLQSGetDUNCallInfo_t Struct Reference</a>	568
8.597.1	Detailed Description	568
8.597.2	Field Documentation	569
8.597.2.1	Mask	569
8.597.2.2	pReportChannelRate	569
8.597.2.3	pReportConnStatus	569
8.597.2.4	pReportDataBearerTech	569
8.597.2.5	pReportDormStatus	569
8.597.2.6	pTransferStatInd	569
8.598	<a href="#">pack_wds_SLQSGetProfileSettings_t Struct Reference</a>	570
8.598.1	Detailed Description	570
8.598.2	Field Documentation	570
8.598.2.1	ProfileId	570
8.598.2.2	ProfileType	570
8.599	<a href="#">pack_wds_SLQSGetRuntimeSettings_t Struct Reference</a>	570
8.599.1	Detailed Description	570
8.599.2	Field Documentation	571
8.599.2.1	pReqSettings	571
8.600	<a href="#">pack_wds_SLQSModifyProfile_t Struct Reference</a>	571
8.600.1	Detailed Description	571
8.600.2	Field Documentation	572
8.600.2.1	curProfile	572
8.600.2.2	pProfileId	572
8.600.2.3	pProfileType	572
8.601	<a href="#">pack_wds_SLQSSet3GPPConfigItem_t Struct Reference</a>	572
8.601.1	Detailed Description	572
8.601.2	Field Documentation	573
8.601.2.1	LTEAttachProfileListLen	573
8.601.2.2	p3gppRelease	573
8.601.2.3	pDefaultPDNEnabled	573
8.601.2.4	pLTEAttachProfile	573
8.601.2.5	pLTEAttachProfileList	573
8.601.2.6	pProfileList	574
8.602	<a href="#">pack_wds_SLQSSetIPFamilyPreference_t Struct Reference</a>	574
8.602.1	Detailed Description	574
8.602.2	Field Documentation	574
8.602.2.1	IPFamilyPreference	574
8.603	<a href="#">pack_wds_SLQSSetWdsEventCallback_t Struct Reference</a>	574

8.603.1 Detailed Description	574
8.603.2 Field Documentation	575
8.603.2.1 currentDataBearer	575
8.603.2.2 dataBearer	575
8.603.2.3 dataBearerTechExt	575
8.603.2.4 dataSystemStatus	575
8.603.2.5 dormancyStatus	575
8.603.2.6 interval	575
8.603.2.7 mobileIP	575
8.603.2.8 transferStats	575
8.604pack_wds_SLQSSGetDHCPv4ClientConfig_t Struct Reference	575
8.604.1 Detailed Description	575
8.604.2 Field Documentation	575
8.604.2.1 pProfileId	575
8.605pack_wds_SLQSSSetDHCPv4ClientConfig_t Struct Reference	575
8.605.1 Detailed Description	575
8.605.2 Field Documentation	576
8.605.2.1 pHwConfig	576
8.605.2.2 pProfileId	576
8.605.2.3 pRequestOptionList	576
8.606pack_wds_SLQSSSetLoopback_t Struct Reference	576
8.606.1 Detailed Description	576
8.606.2 Field Documentation	576
8.606.2.1 loopbackMode	576
8.606.2.2 loopbackMultiplier	576
8.607pack_wds_SLQSSStartDataSession_t Struct Reference	576
8.607.1 Detailed Description	577
8.607.2 Field Documentation	577
8.607.2.1 pAuth	577
8.607.2.2 pPass	577
8.607.2.3 pprofileid3gpp	577
8.607.2.4 pprofileid3gpp2	577
8.607.2.5 pTech	577
8.607.2.6 pUser	578
8.608pack_wds_SLQSSStopDataSession_t Struct Reference	578
8.608.1 Detailed Description	578
8.608.2 Field Documentation	578
8.608.2.1 psid	578
8.609pack_wds_SLQSSwiProfileChangeCallback_t Struct Reference	578
8.609.1 Detailed Description	578

8.609.2 Field Documentation . . . . .	578
8.609.2.1 pProfileChangeInd . . . . .	578
8.610 pack_wds_SLQSWdsSetEventReport_t Struct Reference . . . . .	578
8.610.1 Detailed Description . . . . .	579
8.610.2 Field Documentation . . . . .	580
8.610.2.1 pCurrChannelRateInd . . . . .	580
8.610.2.2 pCurrDataBearerTechInd . . . . .	580
8.610.2.3 pCurrPrefDataSysInd . . . . .	580
8.610.2.4 pDataBearerTechInd . . . . .	580
8.610.2.5 pDataCallStatusChangeInd . . . . .	580
8.610.2.6 pDataSystemStatusChangeInd . . . . .	580
8.610.2.7 pDormancyStatusInd . . . . .	580
8.610.2.8 pEVDOPageMonPerChangeInd . . . . .	580
8.610.2.9 pMIPStatusInd . . . . .	580
8.610.2.10 pTransferStatInd . . . . .	580
8.611 pack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference . . . . .	580
8.611.1 Detailed Description . . . . .	580
8.611.2 Field Documentation . . . . .	581
8.611.2.1 contextId . . . . .	581
8.611.2.2 contextType . . . . .	581
8.612 PackCreateProfileOut Struct Reference . . . . .	581
8.612.1 Detailed Description . . . . .	581
8.612.2 Field Documentation . . . . .	581
8.612.2.1 ExtErrorCode . . . . .	581
8.612.2.2 ProfileIndex . . . . .	581
8.612.2.3 ProfileType . . . . .	581
8.613 packgetDyingGaspCfg Struct Reference . . . . .	581
8.613.1 Detailed Description . . . . .	581
8.613.2 Field Documentation . . . . .	582
8.613.2.1 pDestSMSContent . . . . .	582
8.613.2.2 pDestSMSNum . . . . .	582
8.614 packgetDyingGaspStatistics Struct Reference . . . . .	582
8.614.1 Detailed Description . . . . .	582
8.614.2 Field Documentation . . . . .	582
8.614.2.1 pSMSAttemptedFlag . . . . .	582
8.614.2.2 pTimeStamp . . . . .	582
8.615 PackSwiAvmsSetSettingsAPNInfo Struct Reference . . . . .	582
8.615.1 Detailed Description . . . . .	583
8.615.2 Field Documentation . . . . .	583
8.615.2.1 bAPNLength . . . . .	583

8.615.2.2 bPWDLenght . . . . .	583
8.615.2.3 bUnameLength . . . . .	583
8.615.2.4 szAPN . . . . .	583
8.615.2.5 szPWD . . . . .	583
8.615.2.6 szUname . . . . .	583
8.616PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference . . . . .	583
8.616.1 Detailed Description . . . . .	583
8.616.2 Field Documentation . . . . .	584
8.616.2.1 Timers . . . . .	584
8.617PackSwiAvmsSetSettingsPeriodInfo Struct Reference . . . . .	584
8.617.1 Detailed Description . . . . .	584
8.617.2 Field Documentation . . . . .	584
8.617.2.1 ulMax . . . . .	584
8.617.2.2 ulMin . . . . .	584
8.618PackSwiAVMSSettingsAPNInfo Struct Reference . . . . .	584
8.618.1 Detailed Description . . . . .	585
8.618.2 Field Documentation . . . . .	585
8.618.2.1 bAPNLength . . . . .	585
8.618.2.2 bPWDLenght . . . . .	585
8.618.2.3 bUnameLength . . . . .	585
8.618.2.4 pAPN . . . . .	585
8.618.2.5 pPWD . . . . .	585
8.618.2.6 pUname . . . . .	585
8.619PackSwiAVMSSettingsConnectionRetryTimers Struct Reference . . . . .	585
8.619.1 Detailed Description . . . . .	585
8.619.2 Field Documentation . . . . .	586
8.619.2.1 Timers . . . . .	586
8.620PackSwiAVMSSettingsPeriodsInfo Struct Reference . . . . .	586
8.620.1 Detailed Description . . . . .	586
8.620.2 Field Documentation . . . . .	586
8.620.2.1 max . . . . .	586
8.620.2.2 min . . . . .	586
8.621qmiSmsMessageList Struct Reference . . . . .	586
8.621.1 Detailed Description . . . . .	586
8.621.2 Field Documentation . . . . .	587
8.621.2.1 messageIndex . . . . .	587
8.621.2.2 messageTag . . . . .	587
8.622qmiWSDDataBearerTechnology Struct Reference . . . . .	587
8.622.1 Detailed Description . . . . .	587
8.622.2 Field Documentation . . . . .	587

8.622.2.1 currentNetwork . . . . .	587
8.622.2.2 ratMask . . . . .	587
8.622.2.3 soMask . . . . .	587
8.623qmTlvResult Struct Reference . . . . .	587
8.623.1 Field Documentation . . . . .	588
8.623.1.1 DeviceError . . . . .	588
8.623.1.2 DeviceResult . . . . .	588
8.623.1.3 TlvPresenceMask . . . . .	588
8.623.1.4 TlvResultCode . . . . .	588
8.624qos_BindDataPortMuxID_t Struct Reference . . . . .	588
8.624.1 Detailed Description . . . . .	588
8.624.2 Field Documentation . . . . .	588
8.624.2.1 MuxID . . . . .	588
8.625qos_BindDataPortPeripheralEndPointID_t Struct Reference . . . . .	588
8.625.1 Detailed Description . . . . .	588
8.625.2 Field Documentation . . . . .	589
8.625.2.1 EndPointType . . . . .	589
8.625.2.2 IfaceID . . . . .	589
8.626qos_BindDataPortSIODDataPort_t Struct Reference . . . . .	589
8.626.1 Detailed Description . . . . .	589
8.626.2 Field Documentation . . . . .	589
8.626.2.1 SIODDataPort . . . . .	589
8.627RFBandInfoElements Struct Reference . . . . .	589
8.627.1 Detailed Description . . . . .	589
8.627.2 Field Documentation . . . . .	590
8.627.2.1 activeBandClass . . . . .	590
8.627.2.2 activeChannel . . . . .	590
8.627.2.3 radiolInterface . . . . .	590
8.628rmTrasferStaticsReq Struct Reference . . . . .	590
8.628.1 Detailed Description . . . . .	590
8.628.2 Field Documentation . . . . .	590
8.628.2.1 bResetStatistics . . . . .	590
8.628.2.2 ulMask . . . . .	590
8.629sensorData_t Struct Reference . . . . .	590
8.629.1 Detailed Description . . . . .	591
8.629.2 Field Documentation . . . . .	591
8.629.2.1 flags . . . . .	592
8.629.2.2 sensorDataLen . . . . .	592
8.629.2.3 timeOfFirstSample . . . . .	592
8.629.2.4 timeOffset . . . . .	592

8.629.2.5 xAxis . . . . .	592
8.629.2.6 yAxis . . . . .	592
8.629.2.7 zAxis . . . . .	592
8.630slot_t Struct Reference . . . . .	592
8.630.1 Detailed Description . . . . .	592
8.630.2 Field Documentation . . . . .	593
8.630.2.1 bICCID . . . . .	593
8.630.2.2 bICCIDLength . . . . .	593
8.630.2.3 bLogicalSlot . . . . .	593
8.630.2.4 uPhyCardStatus . . . . .	593
8.630.2.5 uPhySlotStatus . . . . .	593
8.631slotInf Struct Reference . . . . .	593
8.631.1 Detailed Description . . . . .	593
8.631.2 Field Documentation . . . . .	594
8.631.2.1 AppStatus . . . . .	594
8.631.2.2 cardState . . . . .	594
8.631.2.3 errorState . . . . .	594
8.631.2.4 numApp . . . . .	594
8.631.2.5 upinRetries . . . . .	594
8.631.2.6 upinState . . . . .	594
8.631.2.7 upukRetries . . . . .	594
8.632slots_t Struct Reference . . . . .	594
8.632.1 Detailed Description . . . . .	594
8.632.2 Field Documentation . . . . .	595
8.632.2.1 uimSlotStatus . . . . .	595
8.633sms_BroadcastConfig Struct Reference . . . . .	595
8.633.1 Detailed Description . . . . .	595
8.633.2 Field Documentation . . . . .	595
8.633.2.1 fromServiceId . . . . .	595
8.633.2.2 selected . . . . .	595
8.633.2.3 toServiceId . . . . .	595
8.634sms_CDMABroadcastConfig Struct Reference . . . . .	595
8.634.1 Detailed Description . . . . .	596
8.634.2 Field Documentation . . . . .	596
8.634.2.1 language . . . . .	596
8.634.2.2 selected . . . . .	596
8.634.2.3 serviceCategory . . . . .	596
8.635sms_getIndicationReg Struct Reference . . . . .	596
8.635.1 Detailed Description . . . . .	596
8.635.2 Field Documentation . . . . .	597

8.635.2.1 pRegCallStatInfoEvt . . . . .	597
8.635.2.2 pRegTransLayerInfoEvt . . . . .	597
8.635.2.3 pRegTransNWRegInfoEvt . . . . .	597
8.636sms_getMsgWaitingInfo Struct Reference . . . . .	597
8.636.1 Detailed Description . . . . .	597
8.636.2 Field Documentation . . . . .	597
8.636.2.1 msgWaitInfo . . . . .	597
8.636.2.2 numInstances . . . . .	597
8.637sms_getTransLayerInfo Struct Reference . . . . .	597
8.637.1 Detailed Description . . . . .	598
8.637.2 Field Documentation . . . . .	598
8.637.2.1 pRegInd . . . . .	598
8.637.2.2 pTransLayerInfo . . . . .	598
8.638sms_getTransNWRegInfo Struct Reference . . . . .	598
8.638.1 Detailed Description . . . . .	598
8.638.2 Field Documentation . . . . .	598
8.638.2.1 pRegStatus . . . . .	599
8.639sms_maxStorageSizeReq Struct Reference . . . . .	599
8.639.1 Detailed Description . . . . .	599
8.639.2 Field Documentation . . . . .	599
8.639.2.1 pMessageMode . . . . .	599
8.639.2.2 storageType . . . . .	599
8.640sms_maxStorageSizeResp Struct Reference . . . . .	599
8.640.1 Detailed Description . . . . .	599
8.640.2 Field Documentation . . . . .	600
8.640.2.1 freeSlots . . . . .	600
8.640.2.2 maxStorageSize . . . . .	600
8.641sms_messageWaitingInfoContent Struct Reference . . . . .	600
8.641.1 Detailed Description . . . . .	600
8.641.2 Field Documentation . . . . .	600
8.641.2.1 activeInd . . . . .	600
8.641.2.2 msgCount . . . . .	600
8.641.2.3 msgType . . . . .	600
8.642sms_msgProtocolResp Struct Reference . . . . .	601
8.642.1 Detailed Description . . . . .	601
8.642.2 Field Documentation . . . . .	601
8.642.2.1 msgProtocol . . . . .	601
8.643sms_qaQmi3GPP2BroadcastCfgInfo Struct Reference . . . . .	601
8.643.1 Detailed Description . . . . .	601
8.643.2 Field Documentation . . . . .	602



8.643.2.1 activated_ind . . . . .	602
8.643.2.2 CDMABroadcastConfig . . . . .	602
8.643.2.3 num_instances . . . . .	602
8.644sms_qaQmi3GPPBroadcastCfgInfo Struct Reference . . . . .	602
8.644.1 Detailed Description . . . . .	602
8.644.2 Field Documentation . . . . .	602
8.644.2.1 activated_ind . . . . .	602
8.644.2.2 broadcastConfig . . . . .	602
8.644.2.3 num_instances . . . . .	602
8.645sms_routeEntry Struct Reference . . . . .	602
8.645.1 Detailed Description . . . . .	603
8.645.2 Field Documentation . . . . .	603
8.645.2.1 messageClass . . . . .	603
8.645.2.2 messageType . . . . .	603
8.645.2.3 receiptAction . . . . .	603
8.645.2.4 routeStorage . . . . .	603
8.646sms_sendAsyncsmsParams Struct Reference . . . . .	603
8.646.1 Detailed Description . . . . .	604
8.646.2 Field Documentation . . . . .	605
8.646.2.1 messageFormat . . . . .	605
8.646.2.2 messageSize . . . . .	605
8.646.2.3 pFollowOnDC . . . . .	605
8.646.2.4 pForceOnDC . . . . .	605
8.646.2.5 pLinktimer . . . . .	605
8.646.2.6 pMessage . . . . .	605
8.646.2.7 pRetryMessage . . . . .	605
8.646.2.8 pRetryMessageId . . . . .	605
8.646.2.9 pServiceOption . . . . .	605
8.646.2.10pSmsOnIms . . . . .	605
8.646.2.11pUserData . . . . .	605
8.647sms_setIndicationReg Struct Reference . . . . .	606
8.647.1 Detailed Description . . . . .	606
8.647.2 Field Documentation . . . . .	606
8.647.2.1 pRegCallStatInfoEvt . . . . .	606
8.647.2.2 pRegTransLayerInfoEvt . . . . .	606
8.647.2.3 pRegTransNWRegInfoEvt . . . . .	606
8.648sms_setRoutesReq Struct Reference . . . . .	606
8.648.1 Detailed Description . . . . .	607
8.648.2 Field Documentation . . . . .	607
8.648.2.1 numOfRoutes . . . . .	607

8.648.2.2 pTransferStatusReport . . . . .	607
8.648.2.3 routeList . . . . .	607
8.649sms_transLayerInfo Struct Reference . . . . .	607
8.649.1 Detailed Description . . . . .	607
8.649.2 Field Documentation . . . . .	607
8.649.2.1 TransCap . . . . .	608
8.649.2.2 TransType . . . . .	608
8.650sMSCAddressInfo Struct Reference . . . . .	608
8.650.1 Detailed Description . . . . .	608
8.650.2 Field Documentation . . . . .	608
8.650.2.1 data . . . . .	608
8.650.2.2 length . . . . .	608
8.651sMSCAddressTlv Struct Reference . . . . .	608
8.651.1 Detailed Description . . . . .	608
8.651.2 Field Documentation . . . . .	608
8.651.2.1 SMSCInfo . . . . .	609
8.651.2.2 TlvPresent . . . . .	609
8.652sMSEtwsMessageInfo Struct Reference . . . . .	609
8.652.1 Detailed Description . . . . .	609
8.652.2 Field Documentation . . . . .	609
8.652.2.1 data . . . . .	609
8.652.2.2 length . . . . .	609
8.652.2.3 notificationType . . . . .	609
8.653sMSEtwsMessageTlv Struct Reference . . . . .	609
8.653.1 Detailed Description . . . . .	609
8.653.2 Field Documentation . . . . .	610
8.653.2.1 EtwsMessageInfo . . . . .	610
8.653.2.2 TlvPresent . . . . .	610
8.654sMSEtwsPlmnInfo Struct Reference . . . . .	610
8.654.1 Detailed Description . . . . .	610
8.654.2 Field Documentation . . . . .	610
8.654.2.1 mobileCountryCode . . . . .	610
8.654.2.2 mobileNetworkCode . . . . .	610
8.655sMSMessageModeInfo Struct Reference . . . . .	610
8.655.1 Detailed Description . . . . .	610
8.655.2 Field Documentation . . . . .	611
8.655.2.1 messageMode . . . . .	611
8.656sMSMTMessageInfo Struct Reference . . . . .	611
8.656.1 Detailed Description . . . . .	611
8.656.2 Field Documentation . . . . .	611

8.656.2.1 messageIndex . . . . .	611
8.656.2.2 storageType . . . . .	611
8.657sMSOnIMSInfo Struct Reference . . . . .	611
8.657.1 Detailed Description . . . . .	611
8.657.2 Field Documentation . . . . .	611
8.657.2.1 smsOnIMS . . . . .	611
8.658sMSOnIMSTlv Struct Reference . . . . .	612
8.658.1 Detailed Description . . . . .	612
8.658.2 Field Documentation . . . . .	612
8.658.2.1 IMSInfo . . . . .	612
8.658.2.2 TlvPresent . . . . .	612
8.659sMSTransferRouteMTMessageInfo Struct Reference . . . . .	612
8.659.1 Detailed Description . . . . .	612
8.659.2 Field Documentation . . . . .	613
8.659.2.1 ackIndicator . . . . .	613
8.659.2.2 data . . . . .	613
8.659.2.3 format . . . . .	613
8.659.2.4 length . . . . .	613
8.659.2.5 transactionID . . . . .	613
8.660swi_uint256_t Struct Reference . . . . .	613
8.660.1 Detailed Description . . . . .	613
8.660.2 Field Documentation . . . . .	613
8.660.2.1 word . . . . .	613
8.661swiaudio_PCMparams Struct Reference . . . . .	613
8.661.1 Detailed Description . . . . .	613
8.661.2 Field Documentation . . . . .	614
8.661.2.1 iFaceTab . . . . .	614
8.661.2.2 iFaceTabLen . . . . .	614
8.662swidms_ehrpdMTUSizeTlv Struct Reference . . . . .	614
8.662.1 Detailed Description . . . . .	614
8.662.2 Field Documentation . . . . .	614
8.662.2.1 ehrpdMTUSize . . . . .	614
8.662.2.2 TlvPresent . . . . .	614
8.663swidms_hrpdmTUSizeTlv Struct Reference . . . . .	614
8.663.1 Detailed Description . . . . .	614
8.663.2 Field Documentation . . . . .	615
8.663.2.1 hrpdMTUSize . . . . .	615
8.663.2.2 TlvPresent . . . . .	615
8.664swidms_intfaceCfgTlv Struct Reference . . . . .	615
8.664.1 Detailed Description . . . . .	615

8.664.2 Field Documentation	616
8.664.2.1 CfgValue	616
8.664.2.2 CurrentCfgType	616
8.664.2.3 TlvPresent	616
8.665swidms_mtuSize3gppTlv Struct Reference	616
8.665.1 Detailed Description	616
8.665.2 Field Documentation	617
8.665.2.1 MTUSize3gpp	617
8.665.2.2 TlvPresent	617
8.666swidms_supportedIntBitmaskTlv Struct Reference	617
8.666.1 Detailed Description	617
8.666.2 Field Documentation	617
8.666.2.1 TlvPresent	617
8.666.2.2 ValidBitmasks	617
8.667swidms_SwiDmsGetHWWatchdog Struct Reference	617
8.667.1 Detailed Description	617
8.667.2 Field Documentation	618
8.667.2.1 count	618
8.667.2.2 enable	618
8.667.2.3 resetDelay	618
8.667.2.4 timeout	618
8.668swidms_usbMTUSizeTlv Struct Reference	618
8.668.1 Detailed Description	618
8.668.2 Field Documentation	618
8.668.2.1 TlvPresent	618
8.668.2.2 UsbMTUSize	618
8.669tdscdmaSigInfoExt Struct Reference	619
8.669.1 Detailed Description	619
8.669.2 Field Documentation	619
8.669.2.1 ecio	619
8.669.2.2 rscp	619
8.669.2.3 rssi	619
8.669.2.4 sinr	619
8.670tempData_t Struct Reference	619
8.670.1 Detailed Description	619
8.670.2 Field Documentation	620
8.670.2.1 temperature	620
8.670.2.2 temperatureDataLen	620
8.670.2.3 timeOfFirstSample	620
8.670.2.4 timeOffset	620

8.670.2.5 timeSource . . . . .	620
8.671tmd_mitigationDevList Struct Reference . . . . .	620
8.671.1 Detailed Description . . . . .	620
8.671.2 Field Documentation . . . . .	621
8.671.2.1 maxMitigationLevel . . . . .	621
8.671.2.2 mitigationDevId . . . . .	621
8.671.2.3 mitigationDevIdLen . . . . .	621
8.672transferRouteMessageTlv Struct Reference . . . . .	621
8.672.1 Detailed Description . . . . .	621
8.672.2 Field Documentation . . . . .	621
8.672.2.1 TlvPresent . . . . .	621
8.672.2.2 TransferRouteMTMessageInfo . . . . .	621
8.673uim_additionalReadResult Struct Reference . . . . .	621
8.673.1 Detailed Description . . . . .	622
8.673.2 Field Documentation . . . . .	622
8.673.2.1 additionalRecord . . . . .	622
8.673.2.2 additionalRecordLen . . . . .	622
8.673.2.3 TlvPresent . . . . .	622
8.674uim_appStatus Struct Reference . . . . .	622
8.674.1 Detailed Description . . . . .	622
8.674.2 Field Documentation . . . . .	624
8.674.2.1 aidLength . . . . .	624
8.674.2.2 aidVal . . . . .	624
8.674.2.3 appState . . . . .	625
8.674.2.4 appType . . . . .	625
8.674.2.5 persoFeature . . . . .	625
8.674.2.6 persoRetries . . . . .	625
8.674.2.7 persoState . . . . .	625
8.674.2.8 persoUnblockRetries . . . . .	625
8.674.2.9 pin1Retries . . . . .	625
8.674.2.10pin1State . . . . .	625
8.674.2.11pin2Retries . . . . .	625
8.674.2.12pin2State . . . . .	625
8.674.2.13puk1Retries . . . . .	625
8.674.2.14puk2Retries . . . . .	625
8.674.2.15univPin . . . . .	625
8.675uim_authenticateResult Struct Reference . . . . .	625
8.675.1 Detailed Description . . . . .	625
8.675.2 Field Documentation . . . . .	625
8.675.2.1 content . . . . .	625

8.675.2.2 contentLen . . . . .	625
8.676uim_authenticationData Struct Reference . . . . .	626
8.676.1 Detailed Description . . . . .	626
8.676.2 Field Documentation . . . . .	626
8.676.2.1 context . . . . .	626
8.676.2.2 data . . . . .	626
8.676.2.3 dataLen . . . . .	626
8.677uim_cardResult Struct Reference . . . . .	627
8.677.1 Detailed Description . . . . .	627
8.677.2 Field Documentation . . . . .	627
8.677.2.1 sw1 . . . . .	627
8.677.2.2 sw2 . . . . .	627
8.678uim_cardResultInfo Struct Reference . . . . .	627
8.678.1 Detailed Description . . . . .	627
8.678.2 Field Documentation . . . . .	627
8.678.2.1 sw1 . . . . .	628
8.678.2.2 sw2 . . . . .	628
8.678.2.3 TlvPresent . . . . .	628
8.679uim_cardStatus Struct Reference . . . . .	628
8.679.1 Detailed Description . . . . .	628
8.679.2 Field Documentation . . . . .	629
8.679.2.1 index1xPri . . . . .	629
8.679.2.2 index1xSec . . . . .	629
8.679.2.3 indexGwPri . . . . .	629
8.679.2.4 indexGwSec . . . . .	629
8.679.2.5 numSlot . . . . .	629
8.679.2.6 SlotInfo . . . . .	629
8.680uim_changeUIMPIN Struct Reference . . . . .	629
8.680.1 Detailed Description . . . . .	629
8.680.2 Field Documentation . . . . .	629
8.680.2.1 oldPINLen . . . . .	630
8.680.2.2 oldPINVal . . . . .	630
8.680.2.3 pinID . . . . .	630
8.680.2.4 pinLen . . . . .	630
8.680.2.5 pinVal . . . . .	630
8.681uim_depersonalizationInformation Struct Reference . . . . .	630
8.681.1 Detailed Description . . . . .	630
8.681.2 Field Documentation . . . . .	631
8.681.2.1 ckLen . . . . .	631
8.681.2.2 ckVal . . . . .	631

8.681.2.3 feature . . . . .	631
8.681.2.4 operation . . . . .	631
8.682uim_encryptedPIN1 Struct Reference . . . . .	631
8.682.1 Detailed Description . . . . .	631
8.682.2 Field Documentation . . . . .	631
8.682.2.1 pin1Len . . . . .	631
8.682.2.2 pin1Val . . . . .	631
8.683uim_fileAttributes Struct Reference . . . . .	631
8.683.1 Detailed Description . . . . .	632
8.683.2 Field Documentation . . . . .	634
8.683.2.1 fileID . . . . .	634
8.683.2.2 fileSize . . . . .	634
8.683.2.3 fileType . . . . .	634
8.683.2.4 rawLen . . . . .	634
8.683.2.5 rawValue . . . . .	634
8.683.2.6 recordCount . . . . .	634
8.683.2.7 recordSize . . . . .	634
8.683.2.8 secActivate . . . . .	634
8.683.2.9 secActivateMask . . . . .	634
8.683.2.10secDeactivate . . . . .	634
8.683.2.11secDeactivateMask . . . . .	634
8.683.2.12secIncrease . . . . .	634
8.683.2.13secIncreaseMask . . . . .	634
8.683.2.14secRead . . . . .	634
8.683.2.15secReadMask . . . . .	635
8.683.2.16secWrite . . . . .	635
8.683.2.17secWriteMask . . . . .	635
8.684uim_fileInfo Struct Reference . . . . .	635
8.684.1 Detailed Description . . . . .	635
8.684.2 Field Documentation . . . . .	635
8.684.2.1 fileID . . . . .	635
8.684.2.2 path . . . . .	635
8.684.2.3 pathLen . . . . .	635
8.685uim_GetSlotsInfoTlv Struct Reference . . . . .	635
8.685.1 Detailed Description . . . . .	635
8.685.2 Field Documentation . . . . .	636
8.685.2.1 NumberOfPhySlotInfo . . . . .	636
8.685.2.2 TlvPresent . . . . .	636
8.685.2.3 uimSlotInfo . . . . .	636
8.686uim_GetSlotsStatusTlv Struct Reference . . . . .	636

8.686.1 Detailed Description . . . . .	636
8.686.2 Field Documentation . . . . .	636
8.686.2.1 NumberOfPhySlot . . . . .	636
8.686.2.2 TlvPresent . . . . .	636
8.686.2.3 uimSlotStatus . . . . .	636
8.687uim_hotSwapStatus Struct Reference . . . . .	637
8.687.1 Detailed Description . . . . .	637
8.687.2 Field Documentation . . . . .	637
8.687.2.1 hotSwap . . . . .	637
8.687.2.2 hotSwapLength . . . . .	637
8.688uim_indToken Struct Reference . . . . .	637
8.688.1 Detailed Description . . . . .	637
8.688.2 Field Documentation . . . . .	637
8.688.2.1 TlvPresent . . . . .	638
8.688.2.2 token . . . . .	638
8.689uim_personalizationStatus Struct Reference . . . . .	638
8.689.1 Detailed Description . . . . .	638
8.689.2 Field Documentation . . . . .	638
8.689.2.1 feature . . . . .	638
8.689.2.2 numFeatures . . . . .	638
8.689.2.3 unblockLeft . . . . .	638
8.689.2.4 verifyLeft . . . . .	638
8.690uim_physlotInfo Struct Reference . . . . .	639
8.690.1 Detailed Description . . . . .	639
8.690.2 Field Documentation . . . . .	639
8.690.2.1 atrValue . . . . .	639
8.690.2.2 atrValueLen . . . . .	639
8.690.2.3 cardProtocol . . . . .	639
8.690.2.4 iseUICC . . . . .	639
8.690.2.5 numApp . . . . .	639
8.691uim_physlotsInfo Struct Reference . . . . .	639
8.691.1 Detailed Description . . . . .	640
8.691.2 Field Documentation . . . . .	640
8.691.2.1 uimSlotInfo . . . . .	640
8.692uim_readRecordInfo Struct Reference . . . . .	640
8.692.1 Detailed Description . . . . .	640
8.692.2 Field Documentation . . . . .	640
8.692.2.1 length . . . . .	640
8.692.2.2 record . . . . .	640
8.693uim_readResult Struct Reference . . . . .	640



8.693.1 Detailed Description . . . . .	641
8.693.2 Field Documentation . . . . .	641
8.693.2.1 content . . . . .	641
8.693.2.2 contentLen . . . . .	641
8.694uim_readResultInfo Struct Reference . . . . .	641
8.694.1 Detailed Description . . . . .	641
8.694.2 Field Documentation . . . . .	641
8.694.2.1 content . . . . .	641
8.694.2.2 contentLen . . . . .	641
8.694.2.3 TlvPresent . . . . .	641
8.695uim_readTransparentInfo Struct Reference . . . . .	642
8.695.1 Detailed Description . . . . .	642
8.695.2 Field Documentation . . . . .	642
8.695.2.1 length . . . . .	642
8.695.2.2 offset . . . . .	642
8.696uim_refreshevent Struct Reference . . . . .	642
8.696.1 Detailed Description . . . . .	642
8.696.2 Field Documentation . . . . .	643
8.696.2.1 aid . . . . .	643
8.696.2.2 aidLength . . . . .	643
8.696.2.3 arrfileInfo . . . . .	643
8.696.2.4 mode . . . . .	643
8.696.2.5 numOfFiles . . . . .	644
8.696.2.6 sessionType . . . . .	644
8.696.2.7 stage . . . . .	644
8.697uim_registerRefresh Struct Reference . . . . .	644
8.697.1 Detailed Description . . . . .	644
8.697.2 Field Documentation . . . . .	644
8.697.2.1 arrfileInfo . . . . .	644
8.697.2.2 numFiles . . . . .	644
8.697.2.3 registerFlag . . . . .	644
8.697.2.4 voteForInit . . . . .	644
8.698uim_remainingRetries Struct Reference . . . . .	645
8.698.1 Detailed Description . . . . .	645
8.698.2 Field Documentation . . . . .	645
8.698.2.1 unblockLeft . . . . .	645
8.698.2.2 verifyLeft . . . . .	645
8.699uim_sessionInformation Struct Reference . . . . .	645
8.699.1 Detailed Description . . . . .	645
8.699.2 Field Documentation . . . . .	646

8.699.2.1 aid	646
8.699.2.2 aidLength	646
8.699.2.3 sessionType	646
8.700uim_setPINProtection Struct Reference	646
8.700.1 Detailed Description	646
8.700.2 Field Documentation	647
8.700.2.1 pinID	647
8.700.2.2 pinLength	647
8.700.2.3 pinOperation	647
8.700.2.4 pinValue	647
8.701uim_simBusyStatus Struct Reference	647
8.701.1 Detailed Description	647
8.701.2 Field Documentation	647
8.701.2.1 simBusy	647
8.701.2.2 simBusyLength	647
8.702uim_slotInfo Struct Reference	648
8.702.1 Detailed Description	648
8.702.2 Field Documentation	649
8.702.2.1 AppStatus	649
8.702.2.2 cardState	649
8.702.2.3 errorState	649
8.702.2.4 numApp	649
8.702.2.5 upinRetries	649
8.702.2.6 upinState	649
8.702.2.7 upukRetries	649
8.703uim_UIMGetFDNStatus Struct Reference	649
8.703.1 Detailed Description	649
8.703.2 Field Documentation	650
8.703.2.1 FDNStatus	650
8.703.2.2 TlvPresent	650
8.704uim_UIMGetHiddenKeyStatus Struct Reference	650
8.704.1 Detailed Description	650
8.704.2 Field Documentation	650
8.704.2.1 hiddenKey	650
8.704.2.2 TlvPresent	650
8.705uim_UIMGetIndex Struct Reference	650
8.705.1 Detailed Description	650
8.705.2 Field Documentation	651
8.705.2.1 index	651
8.705.2.2 TlvPresent	651

8.706uim_UIMSessionInformation Struct Reference . . . . .	651
8.706.1 Detailed Description . . . . .	651
8.706.2 Field Documentation . . . . .	651
8.706.2.1 aid . . . . .	652
8.706.2.2 aidLength . . . . .	652
8.706.2.3 sessionType . . . . .	652
8.707uim_unblockUIMPIN Struct Reference . . . . .	652
8.707.1 Detailed Description . . . . .	652
8.707.2 Field Documentation . . . . .	652
8.707.2.1 newPINLen . . . . .	652
8.707.2.2 newPINVal . . . . .	652
8.707.2.3 pinID . . . . .	652
8.707.2.4 pukLen . . . . .	652
8.707.2.5 pukVal . . . . .	653
8.708uim_validCardStatus Struct Reference . . . . .	653
8.708.1 Detailed Description . . . . .	653
8.708.2 Field Documentation . . . . .	653
8.708.2.1 validCard . . . . .	653
8.708.2.2 validCardLength . . . . .	653
8.709uim_verifyUIMPIN Struct Reference . . . . .	653
8.709.1 Detailed Description . . . . .	653
8.709.2 Field Documentation . . . . .	654
8.709.2.1 pinID . . . . .	654
8.709.2.2 pinLen . . . . .	654
8.709.2.3 pinVal . . . . .	654
8.710uim_writeRecordInfo Struct Reference . . . . .	654
8.710.1 Detailed Description . . . . .	654
8.710.2 Field Documentation . . . . .	654
8.710.2.1 data . . . . .	654
8.710.2.2 dataLen . . . . .	654
8.710.2.3 record . . . . .	654
8.711uim_writeTransparentInfo Struct Reference . . . . .	654
8.711.1 Detailed Description . . . . .	655
8.711.2 Field Documentation . . . . .	655
8.711.2.1 data . . . . .	655
8.711.2.2 dataLen . . . . .	655
8.711.2.3 offset . . . . .	655
8.712unpack_audio_SLQSGetAudioPathConfig_t Struct Reference . . . . .	655
8.712.1 Detailed Description . . . . .	655
8.712.2 Field Documentation . . . . .	657

8.712.2.1 ParamPresenceMask . . . . .	657
8.712.2.2 pCodecSTGain . . . . .	657
8.712.2.3 pDTMFTXGain . . . . .	657
8.712.2.4 pECMode . . . . .	657
8.712.2.5 pMICGainSelect . . . . .	657
8.712.2.6 pNSEnable . . . . .	657
8.712.2.7 pRXAGCList . . . . .	657
8.712.2.8 pRXAVCAGCSwitch . . . . .	657
8.712.2.9 pRXAVCList . . . . .	657
8.712.2.10pRXPCMIIRFiltr . . . . .	657
8.712.2.11pTXAGCList . . . . .	657
8.712.2.12pTXAVCSwitch . . . . .	657
8.712.2.13pTXGain . . . . .	657
8.712.2.14pTXPCMIIRFiltr . . . . .	657
8.713unpack_audio_SLQSGetAudioProfile_t Struct Reference . . . . .	657
8.713.1 Detailed Description . . . . .	658
8.713.2 Field Documentation . . . . .	658
8.713.2.1 EarMute . . . . .	658
8.713.2.2 MicMute . . . . .	658
8.713.2.3 ParamPresenceMask . . . . .	658
8.713.2.4 Profile . . . . .	658
8.713.2.5 Volume . . . . .	658
8.714unpack_audio_SLQSGetAudioVolTLBConfig_t Struct Reference . . . . .	658
8.714.1 Detailed Description . . . . .	659
8.714.2 Field Documentation . . . . .	659
8.714.2.1 ParamPresenceMask . . . . .	659
8.714.2.2 ResCode . . . . .	659
8.715unpack_audio_SLQSSetAudioVolTLBConfig_t Struct Reference . . . . .	659
8.715.1 Detailed Description . . . . .	659
8.715.2 Field Documentation . . . . .	659
8.715.2.1 ParamPresenceMask . . . . .	659
8.715.2.2 ResCode . . . . .	659
8.716unpack_cat_SetCatEventCallback_ind_t Struct Reference . . . . .	659
8.716.1 Detailed Description . . . . .	660
8.716.2 Field Documentation . . . . .	660
8.716.2.1 CCETiv . . . . .	660
8.716.2.2 event_Index . . . . .	660
8.716.2.3 ParamPresenceMask . . . . .	660
8.717unpack_cat_SetCATEventCallback_t Struct Reference . . . . .	660
8.717.1 Detailed Description . . . . .	660

8.717.2 Field Documentation	661
8.717.2.1 errorMask	661
8.717.2.2 ParamPresenceMask	661
8.717.2.3 Tlvresult	661
8.718unpack_dms_GetActivationState_t Struct Reference	661
8.718.1 Detailed Description	661
8.718.2 Field Documentation	662
8.718.2.1 ParamPresenceMask	662
8.718.2.2 state	662
8.719unpack_dms_GetBandCapability_t Struct Reference	662
8.719.1 Detailed Description	662
8.719.2 Field Documentation	664
8.719.2.1 BandCapability	664
8.719.2.2 ParamPresenceMask	664
8.719.2.3 Tlvresult	664
8.720unpack_dms_GetCrashAction_t Struct Reference	664
8.720.1 Detailed Description	664
8.720.2 Field Documentation	664
8.720.2.1 DevCrashState	664
8.720.2.2 ParamPresenceMask	665
8.720.2.3 Tlvresult	665
8.721unpack_dms_GetCustFeature_t Struct Reference	665
8.721.1 Detailed Description	665
8.721.2 Field Documentation	667
8.721.2.1 DHCPRelayEnabled	667
8.721.2.2 DisableIMSI	667
8.721.2.3 GpsEnable	667
8.721.2.4 GPSLPM	667
8.721.2.5 GPSSel	667
8.721.2.6 IPFamSupport	667
8.721.2.7 IsVoiceEnabled	667
8.721.2.8 ParamPresenceMask	667
8.721.2.9 RMAutoConnect	667
8.721.2.10SMSSupport	667
8.721.2.11Tlvresult	667
8.722unpack_dms_GetCustFeaturesV2_t Struct Reference	667
8.722.1 Detailed Description	667
8.722.2 Field Documentation	668
8.722.2.1 GetCustomFeatureV2	668
8.722.2.2 ParamPresenceMask	668

8.722.2.3 Tlvresult . . . . .	668
8.723unpack_dms_GetDeviceCap_t Struct Reference . . . . .	668
8.723.1 Detailed Description . . . . .	668
8.723.2 Field Documentation . . . . .	669
8.723.2.1 DataServiceCapability . . . . .	669
8.723.2.2 MaxRXChannelRate . . . . .	669
8.723.2.3 MaxTXChannelRate . . . . .	669
8.723.2.4 ParamPresenceMask . . . . .	669
8.723.2.5 Radiolfaces . . . . .	669
8.723.2.6 RadiolfacesSize . . . . .	669
8.723.2.7 SimCapability . . . . .	669
8.723.2.8 Tlvresult . . . . .	669
8.724unpack_dms_GetDeviceCapabilities_t Struct Reference . . . . .	669
8.724.1 Detailed Description . . . . .	670
8.724.2 Field Documentation . . . . .	670
8.724.2.1 dataServiceCaCapability . . . . .	671
8.724.2.2 maxRxChannelRate . . . . .	671
8.724.2.3 maxTxChannelRate . . . . .	671
8.724.2.4 ParamPresenceMask . . . . .	671
8.724.2.5 Radiolfaces . . . . .	671
8.724.2.6 radiolfacesSize . . . . .	671
8.724.2.7 simCapability . . . . .	671
8.725unpack_dms_GetDeviceCapabilitiesV2_t Struct Reference . . . . .	671
8.725.1 Detailed Description . . . . .	671
8.725.2 Field Documentation . . . . .	673
8.725.2.1 DevCaps . . . . .	673
8.725.2.2 ParamPresenceMask . . . . .	673
8.725.2.3 pDevCurSubsCaps . . . . .	673
8.725.2.4 pDevExplicitCfgIndex . . . . .	673
8.725.2.5 pDevMaxActDataSubsCaps . . . . .	673
8.725.2.6 pDevMaxCfgListCaps . . . . .	673
8.725.2.7 pDevMaxSubsCaps . . . . .	673
8.725.2.8 pDevMultiSimCaps . . . . .	673
8.725.2.9 pDevMultiSimVoiceDataCaps . . . . .	673
8.725.2.10pDevSrvCaps . . . . .	673
8.725.2.11pDevSubsFeatureModeCaps . . . . .	673
8.725.2.12pDevSubsVoiceDataCaps . . . . .	673
8.725.2.13pDevVoiceCaps . . . . .	673
8.725.2.14pDevVoiceDataCaps . . . . .	673
8.725.2.15Tlvresult . . . . .	673

8.726unpack_dms_GetDeviceHardwareRev_t Struct Reference . . . . .	673
8.726.1 Detailed Description . . . . .	674
8.726.2 Field Documentation . . . . .	674
8.726.2.1 ParamPresenceMask . . . . .	674
8.726.2.2 String . . . . .	674
8.726.2.3 stringSize . . . . .	674
8.726.2.4 Tlvresult . . . . .	674
8.727unpack_dms_GetDeviceMfr_t Struct Reference . . . . .	674
8.727.1 Detailed Description . . . . .	674
8.727.2 Field Documentation . . . . .	675
8.727.2.1 ParamPresenceMask . . . . .	675
8.727.2.2 String . . . . .	675
8.727.2.3 stringSize . . . . .	675
8.727.2.4 Tlvresult . . . . .	675
8.728unpack_dms_GetDeviceSerialNumbers_t Struct Reference . . . . .	675
8.728.1 Detailed Description . . . . .	675
8.728.2 Field Documentation . . . . .	676
8.728.2.1 esnSize . . . . .	677
8.728.2.2 ESNString . . . . .	677
8.728.2.3 imeiSize . . . . .	677
8.728.2.4 IMEIString . . . . .	677
8.728.2.5 imeiSvnSize . . . . .	677
8.728.2.6 IimeiSvnString . . . . .	677
8.728.2.7 meidSize . . . . .	677
8.728.2.8 MEIDString . . . . .	677
8.728.2.9 ParamPresenceMask . . . . .	677
8.728.2.10Tlvresult . . . . .	677
8.729unpack_dms_GetFirmwareInfo_t Struct Reference . . . . .	677
8.729.1 Detailed Description . . . . .	677
8.729.2 Field Documentation . . . . .	678
8.729.2.1 appversion_str . . . . .	678
8.729.2.2 bootversion_str . . . . .	678
8.729.2.3 carrier_str . . . . .	678
8.729.2.4 cur_carr_name . . . . .	678
8.729.2.5 cur_carr_rev . . . . .	678
8.729.2.6 modelid_str . . . . .	678
8.729.2.7 packageid_str . . . . .	678
8.729.2.8 ParamPresenceMask . . . . .	678
8.729.2.9 priversion_str . . . . .	678
8.729.2.10sku_str . . . . .	678

8.729.2.1 Tlvresult . . . . .	679
8.730unpack_dms_GetFirmwareRevision_t Struct Reference . . . . .	679
8.730.1 Detailed Description . . . . .	679
8.730.2 Field Documentation . . . . .	679
8.730.2.1 amssSize . . . . .	679
8.730.2.2 AMSSString . . . . .	679
8.730.2.3 ParamPresenceMask . . . . .	679
8.730.2.4 PRIString . . . . .	679
8.730.2.5 Tlvresult . . . . .	679
8.731unpack_dms_GetFirmwareRevisions_t Struct Reference . . . . .	680
8.731.1 Detailed Description . . . . .	680
8.731.2 Field Documentation . . . . .	681
8.731.2.1 amssSize . . . . .	681
8.731.2.2 AMSSString . . . . .	681
8.731.2.3 bootSize . . . . .	681
8.731.2.4 BootString . . . . .	681
8.731.2.5 ParamPresenceMask . . . . .	681
8.731.2.6 priSize . . . . .	681
8.731.2.7 PRIString . . . . .	681
8.731.2.8 Tlvresult . . . . .	681
8.732unpack_dms_GetFSN_t Struct Reference . . . . .	681
8.732.1 Detailed Description . . . . .	681
8.732.2 Field Documentation . . . . .	681
8.732.2.1 ParamPresenceMask . . . . .	681
8.732.2.2 String . . . . .	681
8.732.2.3 Tlvresult . . . . .	681
8.733unpack_dms_GetHardwareRevision_t Struct Reference . . . . .	682
8.733.1 Detailed Description . . . . .	682
8.733.2 Field Documentation . . . . .	682
8.733.2.1 hwVer . . . . .	682
8.733.2.2 ParamPresenceMask . . . . .	682
8.734unpack_dms_GetIMSI_t Struct Reference . . . . .	682
8.734.1 Detailed Description . . . . .	682
8.734.2 Field Documentation . . . . .	682
8.734.2.1 imsi . . . . .	683
8.734.2.2 ParamPresenceMask . . . . .	683
8.734.2.3 Tlvresult . . . . .	683
8.735unpack_dms_GetManufacturer_t Struct Reference . . . . .	683
8.735.1 Detailed Description . . . . .	683
8.735.2 Field Documentation . . . . .	683



8.735.2.1 manufacturer . . . . .	683
8.735.2.2 ParamPresenceMask . . . . .	683
8.735.2.3 Tlvresult . . . . .	683
8.736unpack_dms_GetModelID_t Struct Reference . . . . .	683
8.736.1 Detailed Description . . . . .	683
8.736.2 Field Documentation . . . . .	684
8.736.2.1 modelid . . . . .	684
8.736.2.2 ParamPresenceMask . . . . .	684
8.736.2.3 Tlvresult . . . . .	684
8.737unpack_dms_GetNetworkTime_t Struct Reference . . . . .	684
8.737.1 Detailed Description . . . . .	684
8.737.2 Field Documentation . . . . .	685
8.737.2.1 ParamPresenceMask . . . . .	685
8.737.2.2 source . . . . .	685
8.737.2.3 timestamp . . . . .	685
8.737.2.4 Tlvresult . . . . .	685
8.738unpack_dms_GetNetworkTimeV2_t Struct Reference . . . . .	685
8.738.1 Detailed Description . . . . .	685
8.738.2 Field Documentation . . . . .	686
8.738.2.1 ParamPresenceMask . . . . .	686
8.738.2.2 pSysTime . . . . .	686
8.738.2.3 pUsrTime . . . . .	686
8.738.2.4 source . . . . .	686
8.738.2.5 timestamp . . . . .	686
8.738.2.6 Tlvresult . . . . .	686
8.739unpack_dms_GetOfflineReason_t Struct Reference . . . . .	686
8.739.1 Detailed Description . . . . .	686
8.739.2 Field Documentation . . . . .	687
8.739.2.1 ParamPresenceMask . . . . .	687
8.739.2.2 pbPlatform . . . . .	687
8.739.2.3 pReasonMask . . . . .	687
8.739.2.4 Tlvresult . . . . .	687
8.740unpack_dms_GetPower_t Struct Reference . . . . .	687
8.740.1 Detailed Description . . . . .	687
8.740.2 Field Documentation . . . . .	688
8.740.2.1 HardwareControlledMode . . . . .	688
8.740.2.2 OfflineReason . . . . .	688
8.740.2.3 OperationMode . . . . .	688
8.740.2.4 ParamPresenceMask . . . . .	688
8.740.2.5 Tlvresult . . . . .	688

8.741unpack_dms_GetPRLVersion_t Struct Reference	688
8.741.1 Detailed Description	688
8.741.2 Field Documentation	689
8.741.2.1 ParamPresenceMask	689
8.741.2.2 Tlvresult	689
8.741.2.3 u16PRLVersion	689
8.741.2.4 u8PRLPreference	689
8.742unpack_dms_GetSerialNumbers_t Struct Reference	689
8.742.1 Detailed Description	689
8.742.2 Field Documentation	690
8.742.2.1 esn	690
8.742.2.2 imei_no	690
8.742.2.3 imeisv_svn	690
8.742.2.4 meid	690
8.742.2.5 ParamPresenceMask	690
8.743unpack_dms_GetUSBComp_t Struct Reference	690
8.743.1 Detailed Description	690
8.743.2 Field Documentation	692
8.743.2.1 NumSupUSBComps	692
8.743.2.2 ParamPresenceMask	692
8.743.2.3 SupUSBComps	692
8.743.2.4 Tlvresult	692
8.743.2.5 USBComp	692
8.744unpack_dms_GetVoiceNumber_t Struct Reference	692
8.744.1 Detailed Description	693
8.744.2 Field Documentation	693
8.744.2.1 MIN	693
8.744.2.2 minSize	693
8.744.2.3 ParamPresenceMask	693
8.744.2.4 Tlvresult	693
8.744.2.5 VoiceNumber	693
8.744.2.6 voiceNumberSize	693
8.745unpack_dms_PSMCfgChange_ind_t Struct Reference	694
8.745.1 Detailed Description	694
8.745.2 Field Documentation	694
8.745.2.1 ActiveTimer	694
8.745.2.2 EnableState	694
8.745.2.3 ParamPresenceMask	694
8.745.2.4 PeriodicUpdateTimer	694
8.745.2.5 Tlvresult	694

8.746unpack_dms_ResetToFactoryDefaults_t Struct Reference	694
8.746.1 Detailed Description	695
8.746.2 Field Documentation	695
8.746.2.1 ParamPresenceMask	695
8.746.2.2 Tlvresult	695
8.747unpack_dms_SetActivationStatusCallback_t Struct Reference	695
8.747.1 Detailed Description	695
8.747.2 Field Documentation	695
8.747.2.1 ParamPresenceMask	695
8.747.2.2 Tlvresult	695
8.748unpack_dms_SetCrashAction_t Struct Reference	695
8.748.1 Detailed Description	696
8.748.2 Field Documentation	696
8.748.2.1 notused	696
8.748.2.2 ParamPresenceMask	696
8.749unpack_dms_SetCustFeature_t Struct Reference	696
8.749.1 Detailed Description	696
8.749.2 Field Documentation	696
8.749.2.1 ParamPresenceMask	696
8.749.2.2 Tlvresult	696
8.750unpack_dms_SetCustFeaturesV2_t Struct Reference	696
8.750.1 Detailed Description	697
8.750.2 Field Documentation	697
8.750.2.1 ParamPresenceMask	697
8.750.2.2 Tlvresult	697
8.751unpack_dms_SetEventReport_ind_t Struct Reference	697
8.751.1 Detailed Description	697
8.751.2 Field Documentation	697
8.751.2.1 ActivationStatusTlv	697
8.751.2.2 OperatingModeTlv	698
8.751.2.3 ParamPresenceMask	698
8.751.2.4 Tlvresult	698
8.752unpack_dms_SetEventReport_t Struct Reference	698
8.752.1 Detailed Description	698
8.752.2 Field Documentation	698
8.752.2.1 ParamPresenceMask	698
8.752.2.2 Tlvresult	698
8.753unpack_dms_SetFirmwarePreference_t Struct Reference	698
8.753.1 Detailed Description	698
8.753.2 Field Documentation	699

8.753.2.1 ParamPresenceMask . . . . .	699
8.753.2.2 Tlvresult . . . . .	699
8.754unpack_dms_SetIndicationRegister_t Struct Reference . . . . .	699
8.754.1 Detailed Description . . . . .	699
8.754.2 Field Documentation . . . . .	699
8.754.2.1 ParamPresenceMask . . . . .	699
8.754.2.2 Tlvresult . . . . .	699
8.755unpack_dms_SetPower_t Struct Reference . . . . .	699
8.755.1 Detailed Description . . . . .	699
8.755.2 Field Documentation . . . . .	700
8.755.2.1 ParamPresenceMask . . . . .	700
8.755.2.2 Tlvresult . . . . .	700
8.756unpack_dms_SetUSBComp_t Struct Reference . . . . .	700
8.756.1 Detailed Description . . . . .	700
8.756.2 Field Documentation . . . . .	700
8.756.2.1 ParamPresenceMask . . . . .	700
8.756.2.2 Tlvresult . . . . .	700
8.757unpack_dms_SLQSDmsSwiGetPCInfo_t Struct Reference . . . . .	700
8.757.1 Detailed Description . . . . .	700
8.757.2 Field Documentation . . . . .	702
8.757.2.1 has_LpmFlag . . . . .	702
8.757.2.2 has_PersistentLpm . . . . .	702
8.757.2.3 has_PowerOffMode . . . . .	702
8.757.2.4 has_Wdisable . . . . .	703
8.757.2.5 LpmFlag . . . . .	703
8.757.2.6 opMode . . . . .	703
8.757.2.7 ParamPresenceMask . . . . .	703
8.757.2.8 PersistentLpm . . . . .	703
8.757.2.9 PowerOffMode . . . . .	703
8.757.2.10Wdisable . . . . .	703
8.758unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t Struct Reference . . . . .	703
8.758.1 Detailed Description . . . . .	703
8.758.2 Field Documentation . . . . .	704
8.758.2.1 ParamPresenceMask . . . . .	704
8.758.2.2 source . . . . .	704
8.758.2.3 Tlvresult . . . . .	704
8.758.2.4 type . . . . .	704
8.759unpack_dms_SLQSDmsSwiGetResetInfo_t Struct Reference . . . . .	704
8.759.1 Detailed Description . . . . .	704
8.759.2 Field Documentation . . . . .	705

8.759.2.1 ParamPresenceMask . . . . .	705
8.759.2.2 source . . . . .	705
8.759.2.3 Tlvresult . . . . .	705
8.759.2.4 type . . . . .	705
8.760unpack_dms_SLQSDmsSwiGetUimSelection_t Struct Reference . . . . .	705
8.760.1 Detailed Description . . . . .	705
8.760.2 Field Documentation . . . . .	706
8.760.2.1 ParamPresenceMask . . . . .	706
8.760.2.2 pUimAutoSwitchActSlot . . . . .	706
8.760.2.3 uimSelect . . . . .	706
8.761unpack_dms_SLQSDmsSwiIndicationRegister_t Struct Reference . . . . .	706
8.761.1 Detailed Description . . . . .	706
8.761.2 Field Documentation . . . . .	707
8.761.2.1 ParamPresenceMask . . . . .	707
8.761.2.2 Tlvresult . . . . .	707
8.762unpack_dms_SLQSGetBandCapability_t Struct Reference . . . . .	707
8.762.1 Detailed Description . . . . .	707
8.762.2 Field Documentation . . . . .	711
8.762.2.1 bandCapability . . . . .	711
8.762.2.2 is_LteBandCapability_Available . . . . .	711
8.762.2.3 is_TdsBandCapability_Available . . . . .	711
8.762.2.4 LteBandCapability . . . . .	711
8.762.2.5 ParamPresenceMask . . . . .	711
8.762.2.6 TdsBandCapability . . . . .	711
8.763unpack_dms_SLQSGetBandCapabilityExt_t Struct Reference . . . . .	711
8.763.1 Detailed Description . . . . .	711
8.763.2 Field Documentation . . . . .	715
8.763.2.1 bandCapability . . . . .	715
8.763.2.2 is_LteBandCapability_Available . . . . .	715
8.763.2.3 is_TdsBandCapability_Available . . . . .	715
8.763.2.4 LteBandCapability . . . . .	715
8.763.2.5 LteBandsSupport . . . . .	715
8.763.2.6 ParamPresenceMask . . . . .	715
8.763.2.7 TdsBandCapability . . . . .	715
8.764unpack_dms_SLQSGetERIFile_t Struct Reference . . . . .	715
8.764.1 Detailed Description . . . . .	715
8.764.2 Field Documentation . . . . .	716
8.764.2.1 eriFile . . . . .	716
8.764.2.2 ParamPresenceMask . . . . .	716
8.764.2.3 Tlvresult . . . . .	716

8.765unpack_dms_SLQSSGetPowerSaveModeConfig_t Struct Reference	716
8.765.1 Detailed Description	716
8.765.2 Field Documentation	717
8.765.2.1 pActiveTimer	717
8.765.2.2 ParamPresenceMask	717
8.765.2.3 pDurationDueToOOS	717
8.765.2.4 pDurationThreshold	717
8.765.2.5 pEarlyWakeupTime	717
8.765.2.6 pPeriodicUpdateTimer	717
8.765.2.7 pPsmEnableState	717
8.765.2.8 pRandomizationWindow	717
8.766unpack_dms_SLQSSSetPowerSaveModeConfig_t Struct Reference	717
8.766.1 Detailed Description	717
8.766.2 Field Documentation	718
8.766.2.1 ParamPresenceMask	718
8.766.2.2 Tlvresult	718
8.767unpack_dms_SLQSSwiClearDyingGaspStatistics_t Struct Reference	718
8.767.1 Detailed Description	718
8.767.2 Field Documentation	718
8.767.2.1 ParamPresenceMask	718
8.767.2.2 Tlvresult	718
8.768unpack_dms_SLQSSwiGetCrashInfo_t Struct Reference	718
8.768.1 Detailed Description	718
8.768.2 Field Documentation	719
8.768.2.1 crashInfoParam	719
8.768.2.2 ParamPresenceMask	719
8.768.2.3 Tlvresult	719
8.769unpack_dms_SLQSSwiGetDyingGaspCfg_t Struct Reference	719
8.769.1 Detailed Description	719
8.769.2 Field Documentation	720
8.769.2.1 ParamPresenceMask	720
8.769.2.2 pGetDyingGaspCfg	720
8.769.2.3 Tlvresult	720
8.770unpack_dms_SLQSSwiGetDyingGaspStatistics_t Struct Reference	720
8.770.1 Detailed Description	720
8.770.2 Field Documentation	720
8.770.2.1 ParamPresenceMask	720
8.770.2.2 pGetDyingGaspStatistics	720
8.770.2.3 Tlvresult	720
8.771unpack_dms_SLQSSwiGetFirmwareCurr_t Struct Reference	720

8.771.1 Detailed Description	721
8.771.2 Field Documentation	721
8.771.2.1 carrier	721
8.771.2.2 fwvers	721
8.771.2.3 numEntries	721
8.771.2.4 ParamPresenceMask	721
8.771.2.5 pCurrImglInfo	721
8.771.2.6 pkgver	721
8.771.2.7 priver	721
8.772unpack_dms_SLQSSwiGetFwUpdateStatus_t Struct Reference	721
8.772.1 Detailed Description	722
8.772.2 Field Documentation	723
8.772.2.1 imgType	723
8.772.2.2 logString	723
8.772.2.3 ParamPresenceMask	723
8.772.2.4 refData	723
8.772.2.5 refString	723
8.772.2.6 ResCode	723
8.772.2.7 Tlvresult	723
8.773unpack_dms_SLQSSwiGetHostDevInfo_t Struct Reference	723
8.773.1 Detailed Description	723
8.773.2 Field Documentation	724
8.773.2.1 hostID	724
8.773.2.2 manString	724
8.773.2.3 modelString	724
8.773.2.4 ParamPresenceMask	724
8.773.2.5 plasmaIDString	724
8.773.2.6 swVerString	724
8.773.2.7 Tlvresult	724
8.774unpack_dms_SLQSSwiGetOSInfo_t Struct Reference	724
8.774.1 Detailed Description	725
8.774.2 Field Documentation	725
8.774.2.1 nameString	725
8.774.2.2 ParamPresenceMask	725
8.774.2.3 Tlvresult	725
8.774.2.4 versionString	725
8.775unpack_dms_SLQSSwiGetSerialNoExt_t Struct Reference	725
8.775.1 Detailed Description	725
8.775.2 Field Documentation	726
8.775.2.1 meidString	726

8.775.2.2 ParamPresenceMask . . . . .	726
8.775.2.3 Tlvresult . . . . .	726
8.776unpack_dms_SLQSSwiSetDyingGaspCfg_t Struct Reference . . . . .	726
8.776.1 Detailed Description . . . . .	726
8.776.2 Field Documentation . . . . .	726
8.776.2.1 ParamPresenceMask . . . . .	726
8.776.2.2 Tlvresult . . . . .	726
8.777unpack_dms_SLQSSwiSetHostDevInfo_t Struct Reference . . . . .	726
8.777.1 Detailed Description . . . . .	726
8.777.2 Field Documentation . . . . .	727
8.777.2.1 ParamPresenceMask . . . . .	727
8.777.2.2 Tlvresult . . . . .	727
8.778unpack_dms_SLQSSwiSetOSInfo_t Struct Reference . . . . .	727
8.778.1 Detailed Description . . . . .	727
8.778.2 Field Documentation . . . . .	727
8.778.2.1 ParamPresenceMask . . . . .	727
8.778.2.2 Tlvresult . . . . .	727
8.779unpack_dms_SLQSUIIMGetState_t Struct Reference . . . . .	727
8.779.1 Detailed Description . . . . .	727
8.779.2 Field Documentation . . . . .	728
8.779.2.1 ParamPresenceMask . . . . .	728
8.779.2.2 state . . . . .	728
8.779.2.3 Tlvresult . . . . .	728
8.780unpack_dms_SwiEventReportCallBack_ind_t Struct Reference . . . . .	728
8.780.1 Detailed Description . . . . .	728
8.780.2 Field Documentation . . . . .	729
8.780.2.1 ParamPresenceMask . . . . .	729
8.780.2.2 TempTlv . . . . .	729
8.780.2.3 UimStatusTlv . . . . .	729
8.780.2.4 VoltTlv . . . . .	729
8.781unpack_dms_SwiSetEventReport_t Struct Reference . . . . .	729
8.781.1 Detailed Description . . . . .	729
8.781.2 Field Documentation . . . . .	729
8.781.2.1 ParamPresenceMask . . . . .	729
8.781.2.2 Tlvresult . . . . .	729
8.782unpack_dms_SwiUimSelect_t Struct Reference . . . . .	729
8.782.1 Detailed Description . . . . .	729
8.782.2 Field Documentation . . . . .	730
8.782.2.1 ParamPresenceMask . . . . .	730
8.782.2.2 Tlvresult . . . . .	730



8.783unpack_dms_UIMGetControlKeyStatus_t Struct Reference	730
8.783.1 Detailed Description	730
8.783.2 Field Documentation	731
8.783.2.1 facilityState	731
8.783.2.2 ParamPresenceMask	731
8.783.2.3 Tlvresult	731
8.783.2.4 unblockRetriesLeft	731
8.783.2.5 verifyRetriesLeft	731
8.784unpack_dms_UIMGetICCID_t Struct Reference	731
8.784.1 Detailed Description	731
8.784.2 Field Documentation	731
8.784.2.1 ParamPresenceMask	731
8.784.2.2 String	731
8.784.2.3 stringSize	731
8.784.2.4 Tlvresult	731
8.785unpack_dms_UIMGetPINStatus_t Struct Reference	732
8.785.1 Detailed Description	732
8.785.2 Field Documentation	733
8.785.2.1 p1Status	733
8.785.2.2 p1UnblockRetriesLeft	733
8.785.2.3 p1VerifyRetriesLeft	733
8.785.2.4 p2Status	733
8.785.2.5 p2UnblockRetriesLeft	733
8.785.2.6 p2VerifyRetriesLeft	733
8.785.2.7 ParamPresenceMask	733
8.785.2.8 Tlvresult	733
8.786unpack_dms_UIMSetControlKeyProtection_t Struct Reference	733
8.786.1 Detailed Description	733
8.786.2 Field Documentation	734
8.786.2.1 ParamPresenceMask	734
8.786.2.2 Tlvresult	734
8.786.2.3 verifyRetriesLeft	734
8.787unpack_dms_UIMSetPINProtection_t Struct Reference	734
8.787.1 Detailed Description	734
8.787.2 Field Documentation	735
8.787.2.1 ParamPresenceMask	735
8.787.2.2 Tlvresult	735
8.787.2.3 unblockRetriesLeft	735
8.787.2.4 verifyRetriesLeft	735
8.788unpack_dms_UIMUnblockControlKey_t Struct Reference	735

8.788.1 Detailed Description . . . . .	735
8.788.2 Field Documentation . . . . .	735
8.788.2.1 ParamPresenceMask . . . . .	735
8.788.2.2 Tlvresult . . . . .	735
8.788.2.3 unblockRetriesLeft . . . . .	735
8.789unpack_fms_GetImagesPreference_t Struct Reference . . . . .	735
8.789.1 Detailed Description . . . . .	736
8.789.2 Field Documentation . . . . .	736
8.789.2.1 ImageListSize . . . . .	736
8.789.2.2 ParamPresenceMask . . . . .	736
8.789.2.3 plmImageList . . . . .	736
8.789.2.4 Tlvresult . . . . .	736
8.790unpack_fms_GetStoredImages_t Struct Reference . . . . .	736
8.790.1 Detailed Description . . . . .	736
8.790.2 Field Documentation . . . . .	737
8.790.2.1 imageList . . . . .	737
8.790.2.2 imagelistSize . . . . .	737
8.790.2.3 ParamPresenceMask . . . . .	737
8.790.2.4 Tlvresult . . . . .	737
8.791unpack_fms_SetImagesPreference_t Struct Reference . . . . .	737
8.791.1 Detailed Description . . . . .	737
8.791.2 Field Documentation . . . . .	737
8.791.2.1 ImageTypes . . . . .	737
8.791.2.2 ImageTypesSize . . . . .	737
8.791.2.3 ParamPresenceMask . . . . .	737
8.791.2.4 Tlvresult . . . . .	737
8.792unpack_ims_SLQSGetIMSSMSConfig_t Struct Reference . . . . .	737
8.792.1 Detailed Description . . . . .	738
8.792.2 Field Documentation . . . . .	738
8.792.2.1 ParamPresenceMask . . . . .	738
8.792.2.2 pPhoneCtxtURI . . . . .	738
8.792.2.3 pPhoneCtxtURILen . . . . .	738
8.792.2.4 pSettingResp . . . . .	738
8.792.2.5 pSMSFormat . . . . .	738
8.792.2.6 pSMSOverIPNwInd . . . . .	738
8.793unpack_ims_SLQSGetIMSUserConfig_t Struct Reference . . . . .	738
8.793.1 Detailed Description . . . . .	739
8.793.2 Field Documentation . . . . .	739
8.793.2.1 ParamPresenceMask . . . . .	739
8.793.2.2 pIMSDomain . . . . .	739

8.793.2.3 pIMSDomainLen . . . . .	739
8.793.2.4 pSettingResp . . . . .	739
8.794unpack_ims_SLQSGetIMSVoIPConfig_t Struct Reference . . . . .	739
8.794.1 Detailed Description . . . . .	740
8.794.2 Field Documentation . . . . .	741
8.794.2.1 pAmrMode . . . . .	741
8.794.2.2 pAmrOctetAligned . . . . .	741
8.794.2.3 pAmrWbEnable . . . . .	741
8.794.2.4 pAmrWBMode . . . . .	742
8.794.2.5 pAmrWBOctetAligned . . . . .	742
8.794.2.6 ParamPresenceMask . . . . .	742
8.794.2.7 pMinSessionExpiryTimer . . . . .	742
8.794.2.8 pRingBackTimer . . . . .	742
8.794.2.9 pRingingTimer . . . . .	742
8.794.2.10pRTPRTCPInactTimer . . . . .	742
8.794.2.11pScrAmrEnable . . . . .	742
8.794.2.12pScrAmrWbEnable . . . . .	742
8.794.2.13pSessionExpiryTimer . . . . .	742
8.794.2.14pSettingResp . . . . .	742
8.795unpack_ims_SLQSGetRegMgrConfig_t Struct Reference . . . . .	742
8.795.1 Detailed Description . . . . .	742
8.795.2 Field Documentation . . . . .	743
8.795.2.1 ParamPresenceMask . . . . .	743
8.795.2.2 pIMSTestMode . . . . .	743
8.795.2.3 pPCSCFPort . . . . .	743
8.795.2.4 pPriCSCFPortName . . . . .	743
8.795.2.5 pPriCSCFPortNameLen . . . . .	743
8.795.2.6 pSettingResp . . . . .	743
8.796unpack_ims_SLQSGetSIPConfig_t Struct Reference . . . . .	743
8.796.1 Detailed Description . . . . .	743
8.796.2 Field Documentation . . . . .	744
8.796.2.1 ParamPresenceMask . . . . .	744
8.796.2.2 pSettingResp . . . . .	744
8.796.2.3 pSigCompEnabled . . . . .	744
8.796.2.4 pSIPLocalPort . . . . .	744
8.796.2.5 pSubscribeTimer . . . . .	744
8.796.2.6 pTimerSIPReg . . . . .	744
8.796.2.7 pTimerT1 . . . . .	744
8.796.2.8 pTimerT2 . . . . .	744
8.796.2.9 pTimerTf . . . . .	744

8.797unpack_ims_SLQSRegMgrCfgCallBack_ind_t Struct Reference . . . . .	744
8.797.1 Detailed Description . . . . .	745
8.797.2 Field Documentation . . . . .	745
8.797.2.1 ParamPresenceMask . . . . .	745
8.797.2.2 PCTlv . . . . .	745
8.797.2.3 PNTlv . . . . .	745
8.797.2.4 TMTlv . . . . .	745
8.798unpack_ims_SLQSSetIMSSMSConfig_t Struct Reference . . . . .	745
8.798.1 Detailed Description . . . . .	745
8.798.2 Field Documentation . . . . .	746
8.798.2.1 ParamPresenceMask . . . . .	746
8.798.2.2 pSettingResp . . . . .	746
8.799unpack_ims_SLQSSetIMSUserConfig_t Struct Reference . . . . .	746
8.799.1 Detailed Description . . . . .	746
8.799.2 Field Documentation . . . . .	746
8.799.2.1 ParamPresenceMask . . . . .	746
8.799.2.2 pSettingResp . . . . .	746
8.800unpack_ims_SLQSSetIMSVoIPConfig_t Struct Reference . . . . .	746
8.800.1 Detailed Description . . . . .	746
8.800.2 Field Documentation . . . . .	747
8.800.2.1 ParamPresenceMask . . . . .	747
8.800.2.2 pSettingResp . . . . .	747
8.801unpack_ims_SLQSSetRegMgrConfig_t Struct Reference . . . . .	747
8.801.1 Detailed Description . . . . .	747
8.801.2 Field Documentation . . . . .	747
8.801.2.1 ParamPresenceMask . . . . .	747
8.801.2.2 pSettingResp . . . . .	747
8.802unpack_ims_SLQSSetSIPConfig_t Struct Reference . . . . .	747
8.802.1 Detailed Description . . . . .	747
8.802.2 Field Documentation . . . . .	748
8.802.2.1 ParamPresenceMask . . . . .	748
8.802.2.2 pSettingResp . . . . .	748
8.803unpack_ims_SLQSSIPCfgCallBack_ind_t Struct Reference . . . . .	748
8.803.1 Detailed Description . . . . .	748
8.803.2 Field Documentation . . . . .	749
8.803.2.1 ParamPresenceMask . . . . .	749
8.803.2.2 SCTlv . . . . .	749
8.803.2.3 SPTlv . . . . .	749
8.803.2.4 SRTlv . . . . .	749
8.803.2.5 STTlv . . . . .	749

8.803.2.6 TT1Tlv . . . . .	749
8.803.2.7 TT2Tlv . . . . .	749
8.803.2.8 TTfTlv . . . . .	749
8.804unpack_ims_SLQSSMSCfgCallBack_ind_t Struct Reference . . . . .	749
8.804.1 Detailed Description . . . . .	749
8.804.2 Field Documentation . . . . .	750
8.804.2.1 ParamPresenceMask . . . . .	750
8.804.2.2 PCURTlv . . . . .	750
8.804.2.3 SFTlv . . . . .	750
8.804.2.4 SINTlv . . . . .	750
8.805unpack_ims_SLQSUserCfgCallBack_ind_t Struct Reference . . . . .	750
8.805.1 Detailed Description . . . . .	750
8.805.2 Field Documentation . . . . .	751
8.805.2.1 IDTlv . . . . .	751
8.805.2.2 ParamPresenceMask . . . . .	751
8.806unpack_ims_SLQSVolPCfgCallBack_ind_t Struct Reference . . . . .	751
8.806.1 Detailed Description . . . . .	751
8.806.2 Field Documentation . . . . .	752
8.806.2.1 AMTlv . . . . .	752
8.806.2.2 AOATlv . . . . .	752
8.806.2.3 AWMTlv . . . . .	752
8.806.2.4 AWOATlv . . . . .	753
8.806.2.5 EAWTlv . . . . .	753
8.806.2.6 ESATlv . . . . .	753
8.806.2.7 ESAWTlv . . . . .	753
8.806.2.8 MSETlv . . . . .	753
8.806.2.9 ParamPresenceMask . . . . .	753
8.806.2.10RBTTlv . . . . .	753
8.806.2.11RTIDTlv . . . . .	753
8.806.2.12RTTlv . . . . .	753
8.806.2.13SDTlv . . . . .	753
8.807unpack_imsa_SLQSGetIMSARegStatus_t Struct Reference . . . . .	753
8.807.1 Detailed Description . . . . .	753
8.807.2 Field Documentation . . . . .	754
8.807.2.1 ImsRegErrCode . . . . .	754
8.807.2.2 ImsRegStatus . . . . .	754
8.807.2.3 NewImsRegStatus . . . . .	754
8.807.2.4 ParamPresenceMask . . . . .	754
8.808unpack_imsa_SLQSGetIMSAServiceStatus_t Struct Reference . . . . .	754
8.808.1 Detailed Description . . . . .	754

8.808.2 Field Documentation . . . . .	756
8.808.2.1 ParamPresenceMask . . . . .	756
8.808.2.2 SmsServiceRat . . . . .	756
8.808.2.3 SmsServiceStatus . . . . .	756
8.808.2.4 UtServiceRat . . . . .	756
8.808.2.5 UtServiceStatus . . . . .	756
8.808.2.6 VoipServiceRat . . . . .	756
8.808.2.7 VoipServiceStatus . . . . .	756
8.808.2.8 VsServiceRat . . . . .	756
8.808.2.9 VsServiceStatus . . . . .	756
8.808.2.10VtServiceRat . . . . .	756
8.808.2.11VtServiceStatus . . . . .	756
8.809unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t Struct Reference . . . . .	756
8.809.1 Detailed Description . . . . .	757
8.809.2 Field Documentation . . . . .	757
8.809.2.1 FailErrCode . . . . .	757
8.809.2.2 ParamPresenceMask . . . . .	757
8.809.2.3 PdpConnState . . . . .	757
8.810unpack_imsa_SLQSImsaRatStatusCallBack_ind_t Struct Reference . . . . .	757
8.810.1 Detailed Description . . . . .	757
8.810.2 Field Documentation . . . . .	758
8.810.2.1 ParamPresenceMask . . . . .	758
8.810.2.2 RatHandover . . . . .	758
8.811unpack_imsa_SLQSImsaRegStatusCallBack_ind_t Struct Reference . . . . .	758
8.811.1 Detailed Description . . . . .	758
8.811.2 Field Documentation . . . . .	758
8.811.2.1 IMSRegistration . . . . .	758
8.811.2.2 IMSRegistrationError . . . . .	758
8.811.2.3 NewIMSRegistration . . . . .	758
8.811.2.4 ParamPresenceMask . . . . .	759
8.812unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t Struct Reference . . . . .	759
8.812.1 Detailed Description . . . . .	759
8.812.2 Field Documentation . . . . .	760
8.812.2.1 ParamPresenceMask . . . . .	760
8.812.2.2 SmsRat . . . . .	760
8.812.2.3 SmsService . . . . .	760
8.812.2.4 UtRat . . . . .	760
8.812.2.5 UtService . . . . .	760
8.812.2.6 VoipRat . . . . .	760
8.812.2.7 VoipService . . . . .	760

8.812.2.8 VtRat . . . . .	760
8.812.2.9 VtService . . . . .	760
8.813unpack_loc_BestAvailPos_Ind_t Struct Reference . . . . .	760
8.813.1 Detailed Description . . . . .	761
8.813.2 Field Documentation . . . . .	766
8.813.2.1 pAltitudeWrtEllipsoid . . . . .	766
8.813.2.2 pAltitudeWrtMeanSeaLevel . . . . .	766
8.813.2.3 ParamPresenceMask . . . . .	766
8.813.2.4 pGpsTime . . . . .	766
8.813.2.5 pHeading . . . . .	766
8.813.2.6 pHeadingUnc . . . . .	766
8.813.2.7 pHorCirConf . . . . .	766
8.813.2.8 pHorEllpConf . . . . .	766
8.813.2.9 pHorReliability . . . . .	766
8.813.2.10pHorUncCircular . . . . .	766
8.813.2.11pHorUncEllipseOrientAzimuth . . . . .	766
8.813.2.12pHorUncEllipseSemiMajor . . . . .	766
8.813.2.13pHorUncEllipseSemiMinor . . . . .	766
8.813.2.14pLatitude . . . . .	766
8.813.2.15pLongitude . . . . .	766
8.813.2.16pMagneticDeviation . . . . .	766
8.813.2.17pPrecisionDilution . . . . .	766
8.813.2.18pSensorDataUsage . . . . .	766
8.813.2.19pSpeedHorizontal . . . . .	766
8.813.2.20pSpeedUnc . . . . .	766
8.813.2.21pSpeedVertical . . . . .	766
8.813.2.22pSpeedVerticalUnc . . . . .	766
8.813.2.23pSvUsedforFix . . . . .	766
8.813.2.24pTechnologyMask . . . . .	766
8.813.2.25pTimeSrc . . . . .	766
8.813.2.26pTimestampUtc . . . . .	766
8.813.2.27pTimeUnc . . . . .	766
8.813.2.28pVertConfidence . . . . .	767
8.813.2.29pVertReliability . . . . .	767
8.813.2.30pVertUnc . . . . .	767
8.813.2.31pXid . . . . .	767
8.813.2.32status . . . . .	767
8.813.2.33Tlvresult . . . . .	767
8.814unpack_loc_CradleMountCallback_Ind_t Struct Reference . . . . .	767
8.814.1 Detailed Description . . . . .	767

8.814.2 Field Documentation . . . . .	767
8.814.2.1 cradleMountConfigStatus . . . . .	767
8.814.2.2 ParamPresenceMask . . . . .	767
8.815unpack_loc_Delete_Assist_Data_t Struct Reference . . . . .	767
8.815.1 Detailed Description . . . . .	768
8.815.2 Field Documentation . . . . .	768
8.815.2.1 ParamPresenceMask . . . . .	768
8.815.2.2 Tlvresult . . . . .	768
8.816unpack_loc_DeleteAssistData_Ind_t Struct Reference . . . . .	768
8.816.1 Detailed Description . . . . .	768
8.816.2 Field Documentation . . . . .	769
8.816.2.1 ParamPresenceMask . . . . .	769
8.816.2.2 status . . . . .	769
8.816.2.3 Tlvresult . . . . .	769
8.817unpack_loc_EngineState_Ind_t Struct Reference . . . . .	769
8.817.1 Detailed Description . . . . .	769
8.817.2 Field Documentation . . . . .	770
8.817.2.1 engineState . . . . .	770
8.817.2.2 ParamPresenceMask . . . . .	770
8.817.2.3 Tlvresult . . . . .	770
8.818unpack_loc_EventNMEA_Ind_t Struct Reference . . . . .	770
8.818.1 Detailed Description . . . . .	770
8.818.2 Field Documentation . . . . .	770
8.818.2.1 NMEADData . . . . .	770
8.818.2.2 ParamPresenceMask . . . . .	770
8.818.2.3 Tlvresult . . . . .	770
8.819unpack_loc_EventRegister_t Struct Reference . . . . .	771
8.819.1 Detailed Description . . . . .	771
8.819.2 Field Documentation . . . . .	771
8.819.2.1 ParamPresenceMask . . . . .	771
8.819.2.2 Tlvresult . . . . .	771
8.820unpack_loc_EventTimeSyncCallback_Ind_t Struct Reference . . . . .	771
8.820.1 Detailed Description . . . . .	771
8.820.2 Field Documentation . . . . .	771
8.820.2.1 ParamPresenceMask . . . . .	771
8.820.2.2 timeSyncRefCount . . . . .	771
8.821unpack_loc_FixCriteria_Ind_t Struct Reference . . . . .	771
8.821.1 Detailed Description . . . . .	772
8.821.2 Field Documentation . . . . .	772
8.821.2.1 loc_appProviderInfo . . . . .	772



8.821.2.2 loc_fixCriteriaStatus . . . . .	772
8.821.2.3 loc_horAccuracy . . . . .	772
8.821.2.4 loc_intermediateRptState . . . . .	772
8.821.2.5 loc_minInterval . . . . .	772
8.821.2.6 ParamPresenceMask . . . . .	772
8.822unpack_loc_GetOpMode_Ind_t Struct Reference . . . . .	773
8.822.1 Detailed Description . . . . .	773
8.822.2 Field Documentation . . . . .	774
8.822.2.1 ParamPresenceMask . . . . .	774
8.822.2.2 pMode . . . . .	774
8.822.2.3 Status . . . . .	774
8.823unpack_loc_GetServer_Ind_t Struct Reference . . . . .	774
8.823.1 Detailed Description . . . . .	774
8.823.2 Field Documentation . . . . .	775
8.823.2.1 ParamPresenceMask . . . . .	775
8.823.2.2 pIPv4AddrInfo . . . . .	775
8.823.2.3 pIPv6AddrInfo . . . . .	775
8.823.2.4 pURL . . . . .	775
8.823.2.5 serverStatus . . . . .	775
8.823.2.6 serverType . . . . .	775
8.823.2.7 Tlvresult . . . . .	775
8.824unpack_loc_GnssSvInfo_Ind_t Struct Reference . . . . .	775
8.824.1 Detailed Description . . . . .	775
8.824.2 Field Documentation . . . . .	776
8.824.2.1 altitudeAssumed . . . . .	776
8.824.2.2 ParamPresenceMask . . . . .	776
8.824.2.3 pSatelliteInfo . . . . .	776
8.824.2.4 Tlvresult . . . . .	776
8.825unpack_loc_InjectPositionCallback_Ind_t Struct Reference . . . . .	776
8.825.1 Detailed Description . . . . .	776
8.825.2 Field Documentation . . . . .	777
8.825.2.1 ParamPresenceMask . . . . .	777
8.825.2.2 status . . . . .	777
8.826unpack_loc_InjectSensorDataCallback_Ind_t Struct Reference . . . . .	777
8.826.1 Detailed Description . . . . .	777
8.826.2 Field Documentation . . . . .	778
8.826.2.1 injectSensorDataStatus . . . . .	778
8.826.2.2 pAccelSamplesAccepted . . . . .	778
8.826.2.3 pAccelTempSamplesAccepted . . . . .	779
8.826.2.4 ParamPresenceMask . . . . .	779

8.826.2.5 pGyroSamplesAccepted . . . . .	779
8.826.2.6 pGyroTempSamplesAccepted . . . . .	779
8.826.2.7 pOpaqueldentifier . . . . .	779
8.827unpack_loc_InjectTimeSyncDataCallback_Ind_t Struct Reference . . . . .	779
8.827.1 Detailed Description . . . . .	779
8.827.2 Field Documentation . . . . .	779
8.827.2.1 injectTimeSyncStatus . . . . .	779
8.827.2.2 ParamPresenceMask . . . . .	779
8.828unpack_loc_InjectUTCTimeCallback_Ind_t Struct Reference . . . . .	779
8.828.1 Detailed Description . . . . .	780
8.828.2 Field Documentation . . . . .	780
8.828.2.1 ParamPresenceMask . . . . .	780
8.828.2.2 status . . . . .	780
8.829unpack_loc_PositionRpt_Ind_t Struct Reference . . . . .	780
8.829.1 Detailed Description . . . . .	781
8.829.2 Field Documentation . . . . .	786
8.829.2.1 pAltitudeAssumed . . . . .	786
8.829.2.2 pAltitudeWrtEllipsoid . . . . .	786
8.829.2.3 pAltitudeWrtMeanSeaLevel . . . . .	786
8.829.2.4 ParamPresenceMask . . . . .	786
8.829.2.5 pFixId . . . . .	786
8.829.2.6 pGpsTime . . . . .	786
8.829.2.7 pHeading . . . . .	786
8.829.2.8 pHeadingUnc . . . . .	786
8.829.2.9 pHorConfidence . . . . .	786
8.829.2.10pHorReliability . . . . .	786
8.829.2.11pHorUncCircular . . . . .	786
8.829.2.12pHorUncEllipseOrientAzimuth . . . . .	786
8.829.2.13pHorUncEllipseSemiMajor . . . . .	786
8.829.2.14pHorUncEllipseSemiMinor . . . . .	786
8.829.2.15pLatitude . . . . .	786
8.829.2.16pLeapSeconds . . . . .	786
8.829.2.17pLongitude . . . . .	786
8.829.2.18pMagneticDeviation . . . . .	786
8.829.2.19pPrecisionDilution . . . . .	786
8.829.2.20pSensorDataUsage . . . . .	786
8.829.2.21pSpeedHorizontal . . . . .	786
8.829.2.22pSpeedUnc . . . . .	786
8.829.2.23pSpeedVertical . . . . .	786
8.829.2.24pSvUsedforFix . . . . .	786

8.829.2.25pTechnologyMask . . . . .	786
8.829.2.26pTimeSrc . . . . .	786
8.829.2.27pTimestampUtc . . . . .	786
8.829.2.28pTimeUnc . . . . .	787
8.829.2.29pVertConfidence . . . . .	787
8.829.2.30pVertReliability . . . . .	787
8.829.2.31pVertUnc . . . . .	787
8.829.2.32sessionId . . . . .	787
8.829.2.33sessionStatus . . . . .	787
8.829.2.34Tlvresult . . . . .	787
8.830unpack_loc_SensorStreamingCallback_Ind_t Struct Reference . . . . .	787
8.830.1 Detailed Description . . . . .	787
8.830.2 Field Documentation . . . . .	787
8.830.2.1 pAccelAcceptReady . . . . .	787
8.830.2.2 pAccelTempAcceptReady . . . . .	787
8.830.2.3 ParamPresenceMask . . . . .	788
8.830.2.4 pGyroAcceptReady . . . . .	788
8.830.2.5 pGyroTempAcceptReady . . . . .	788
8.831unpack_loc_SetExtPowerConfig_Ind_t Struct Reference . . . . .	788
8.831.1 Detailed Description . . . . .	788
8.831.2 Field Documentation . . . . .	788
8.831.2.1 ParamPresenceMask . . . . .	788
8.831.2.2 status . . . . .	788
8.831.2.3 Tlvresult . . . . .	788
8.832unpack_loc_SetExtPowerState_t Struct Reference . . . . .	788
8.832.1 Detailed Description . . . . .	789
8.832.2 Field Documentation . . . . .	789
8.832.2.1 ParamPresenceMask . . . . .	789
8.832.2.2 Tlvresult . . . . .	789
8.833unpack_loc_SetOperationMode_Ind_t Struct Reference . . . . .	789
8.833.1 Detailed Description . . . . .	789
8.833.2 Field Documentation . . . . .	790
8.833.2.1 ParamPresenceMask . . . . .	790
8.833.2.2 status . . . . .	790
8.833.2.3 Tlvresult . . . . .	790
8.834unpack_loc_SetOperationMode_t Struct Reference . . . . .	790
8.834.1 Detailed Description . . . . .	790
8.834.2 Field Documentation . . . . .	790
8.834.2.1 ParamPresenceMask . . . . .	790
8.834.2.2 Tlvresult . . . . .	790

8.835unpack_loc_SetServer_Ind_t Struct Reference . . . . .	790
8.835.1 Detailed Description . . . . .	790
8.835.2 Field Documentation . . . . .	791
8.835.2.1 ParamPresenceMask . . . . .	791
8.835.2.2 serverStatus . . . . .	791
8.835.2.3 Tlvresult . . . . .	791
8.836unpack_loc_SLQSLOCGetBestAvailPos_t Struct Reference . . . . .	791
8.836.1 Detailed Description . . . . .	791
8.836.2 Field Documentation . . . . .	792
8.836.2.1 ParamPresenceMask . . . . .	792
8.836.2.2 Tlvresult . . . . .	792
8.837unpack_loc_SLQSLOCGetOpMode_t Struct Reference . . . . .	792
8.837.1 Detailed Description . . . . .	792
8.837.2 Field Documentation . . . . .	792
8.837.2.1 ParamPresenceMask . . . . .	792
8.837.2.2 Tlvresult . . . . .	792
8.838unpack_loc_Start_t Struct Reference . . . . .	792
8.838.1 Detailed Description . . . . .	792
8.838.2 Field Documentation . . . . .	793
8.838.2.1 ParamPresenceMask . . . . .	793
8.838.2.2 Tlvresult . . . . .	793
8.839unpack_loc_Stop_t Struct Reference . . . . .	793
8.839.1 Detailed Description . . . . .	793
8.839.2 Field Documentation . . . . .	793
8.839.2.1 ParamPresenceMask . . . . .	793
8.839.2.2 Tlvresult . . . . .	793
8.840unpack_nas_GetACCOLC_t Struct Reference . . . . .	793
8.840.1 Detailed Description . . . . .	793
8.840.2 Field Documentation . . . . .	794
8.840.2.1 pAccolc . . . . .	794
8.840.2.2 ParamPresenceMask . . . . .	794
8.841unpack_nas_GetANAAAuthenticationStatus_t Struct Reference . . . . .	794
8.841.1 Detailed Description . . . . .	794
8.841.2 Field Documentation . . . . .	794
8.841.2.1 ParamPresenceMask . . . . .	794
8.841.2.2 pAuthStatus . . . . .	794
8.842unpack_nas_GetCDMANetworkParameters_t Struct Reference . . . . .	794
8.842.1 Detailed Description . . . . .	795
8.842.2 Field Documentation . . . . .	796
8.842.2.1 Application . . . . .	796

8.842.2.2 Broadcast	796
8.842.2.3 CustomSCP	796
8.842.2.4 ForceRev0	796
8.842.2.5 ParamPresenceMask	796
8.842.2.6 Protocol	796
8.842.2.7 RegForeignNID	796
8.842.2.8 RegForeignSID	797
8.842.2.9 RegHomeSID	797
8.842.2.10Roaming	797
8.842.2.11SCI	797
8.842.2.12SCM	797
8.843unpack_nas_GetHomeNetwork3GPP2_t Struct Reference	797
8.843.1 Detailed Description	797
8.843.2 Field Documentation	799
8.843.2.1 nameSize	799
8.843.2.2 ParamPresenceMask	799
8.843.2.3 pMCC	799
8.843.2.4 pMNC	799
8.843.2.5 pName	799
8.843.2.6 pNID	799
8.843.2.7 pNw2DescDisp	799
8.843.2.8 pNw2DescEnc	799
8.843.2.9 pNw2DescLen	799
8.843.2.10pNw2MCC	799
8.843.2.11pNw2MNC	799
8.843.2.12pNw2Name	799
8.843.2.13pSID	799
8.844unpack_nas_GetHomeNetwork_t Struct Reference	799
8.844.1 Detailed Description	799
8.844.2 Field Documentation	800
8.844.2.1 mcc	800
8.844.2.2 mnc	800
8.844.2.3 name	800
8.844.2.4 nid	800
8.844.2.5 ParamPresenceMask	800
8.844.2.6 sid	800
8.845unpack_nas_GetNetworkPreference_t Struct Reference	800
8.845.1 Detailed Description	800
8.845.2 Field Documentation	801
8.845.2.1 ActiveTechPref	801

8.845.2.2 Duration . . . . .	801
8.845.2.3 ParamPresenceMask . . . . .	801
8.845.2.4 PersistentTechPref . . . . .	801
8.845.2.5 Tlvresult . . . . .	801
8.846unpack_nas_GetRFInfo_t Struct Reference . . . . .	801
8.846.1 Detailed Description . . . . .	802
8.846.2 Field Documentation . . . . .	802
8.846.2.1 instancesSize . . . . .	802
8.846.2.2 ParamPresenceMask . . . . .	802
8.846.2.3 RFBandInfoElements . . . . .	802
8.847unpack_nas_GetServingNetwork_t Struct Reference . . . . .	802
8.847.1 Detailed Description . . . . .	803
8.847.2 Field Documentation . . . . .	804
8.847.2.1 CSDomain . . . . .	804
8.847.2.2 DataCaps . . . . .	804
8.847.2.3 DataCapsLen . . . . .	804
8.847.2.4 MCC . . . . .	804
8.847.2.5 MNC . . . . .	804
8.847.2.6 Name . . . . .	805
8.847.2.7 nameSize . . . . .	805
8.847.2.8 ParamPresenceMask . . . . .	805
8.847.2.9 PSDomain . . . . .	805
8.847.2.10Radiolfaces . . . . .	805
8.847.2.11RadiolfacesSize . . . . .	805
8.847.2.12RAN . . . . .	805
8.847.2.13RegistrationState . . . . .	805
8.847.2.14Roaming . . . . .	805
8.848unpack_nas_GetServingNetworkCapabilities_t Struct Reference . . . . .	805
8.848.1 Detailed Description . . . . .	805
8.848.2 Field Documentation . . . . .	806
8.848.2.1 DataCaps . . . . .	806
8.848.2.2 DataCapsLen . . . . .	806
8.848.2.3 ParamPresenceMask . . . . .	806
8.849unpack_nas_GetSignalStrengths_t Struct Reference . . . . .	806
8.849.1 Detailed Description . . . . .	806
8.849.2 Field Documentation . . . . .	806
8.849.2.1 len . . . . .	806
8.849.2.2 ParamPresenceMask . . . . .	806
8.849.2.3 radio . . . . .	806
8.849.2.4 rssi . . . . .	806

8.850unpack_nas_PerformNetworkScan_t Struct Reference	807
8.850.1 Detailed Description	807
8.850.2 Field Documentation	808
8.850.2.1 p3GppNetworkInfoInstances	808
8.850.2.2 p3GppNetworkInstanceSize	808
8.850.2.3 ParamPresenceMask	808
8.850.2.4 pLteOpModeTlv	808
8.850.2.5 pPCIInfo	808
8.850.2.6 pPCSInstance	808
8.850.2.7 pPCSInstanceSize	808
8.850.2.8 pRATInstance	808
8.850.2.9 pRATInstanceSize	808
8.850.2.10pScanResult	808
8.851unpack_nas_SetDataCapabilitiesCallback_ind_t Struct Reference	808
8.851.1 Detailed Description	809
8.851.2 Field Documentation	809
8.851.2.1 dataCaps	809
8.851.2.2 dataCapsSize	809
8.851.2.3 ParamPresenceMask	809
8.852unpack_nas_SetEventReportInd_t Struct Reference	809
8.852.1 Detailed Description	809
8.852.2 Field Documentation	810
8.852.2.1 ParamPresenceMask	810
8.852.2.2 RFTlv	810
8.852.2.3 RRTlv	810
8.852.2.4 SLQSSSTlv	810
8.852.2.5 SSTlv	810
8.853unpack_nas_SetNasLTECphyCaIndCallback_ind_t Struct Reference	810
8.853.1 Detailed Description	810
8.853.2 Field Documentation	811
8.853.2.1 ParamPresenceMask	811
8.853.2.2 sNumScellsConfig	811
8.853.2.3 sPhyCaAggDIBW	811
8.853.2.4 sPhyCaAggPcellInfo	811
8.853.2.5 sPhyCaAggScellDIBw	811
8.853.2.6 sPhyCaAggScellIndex	811
8.853.2.7 sPhyCaAggScellIndType	811
8.853.2.8 sPhyCaAggScellInfo	811
8.854unpack_nas_SetNetworkPreference_t Struct Reference	811
8.854.1 Detailed Description	812

8.854.2 Field Documentation . . . . .	812
8.854.2.1 ParamPresenceMask . . . . .	812
8.854.2.2 Tlvresult . . . . .	812
8.855unpack_nas_SetRoamingIndicatorCallback_ind_t Struct Reference . . . . .	812
8.855.1 Detailed Description . . . . .	812
8.855.2 Field Documentation . . . . .	812
8.855.2.1 ParamPresenceMask . . . . .	812
8.855.2.2 roaming . . . . .	812
8.856unpack_nas_SetServingSystemCallback_ind_t Struct Reference . . . . .	812
8.856.1 Detailed Description . . . . .	813
8.856.2 Field Documentation . . . . .	813
8.856.2.1 ParamPresenceMask . . . . .	813
8.856.2.2 SSInfo . . . . .	813
8.856.2.3 Tlvresult . . . . .	813
8.857unpack_nas_SLQSGetErrorRate_t Struct Reference . . . . .	813
8.857.1 Detailed Description . . . . .	813
8.857.2 Field Documentation . . . . .	814
8.857.2.1 ParamPresenceMask . . . . .	814
8.857.2.2 pCDMAFrameErrRate . . . . .	814
8.857.2.3 pGSMBER . . . . .	814
8.857.2.4 pHDRPackErrRate . . . . .	814
8.857.2.5 pWCDMABER . . . . .	814
8.858unpack_nas_SLQSGetHomeNetwork_t Struct Reference . . . . .	814
8.858.1 Detailed Description . . . . .	815
8.858.2 Field Documentation . . . . .	815
8.858.2.1 mcc . . . . .	816
8.858.2.2 mnc . . . . .	816
8.858.2.3 name . . . . .	816
8.858.2.4 nid . . . . .	816
8.858.2.5 ParamPresenceMask . . . . .	816
8.858.2.6 pHomeNwMNC3Gpp . . . . .	816
8.858.2.7 pNwNameSrc3Gpp . . . . .	816
8.858.2.8 sid . . . . .	816
8.859unpack_nas_SlqsGetLTECphyCAInfo_t Struct Reference . . . . .	816
8.859.1 Detailed Description . . . . .	816
8.859.2 Field Documentation . . . . .	816
8.859.2.1 LTECphyCAInfo . . . . .	816
8.859.2.2 ParamPresenceMask . . . . .	816
8.859.2.3 Tlvresult . . . . .	816
8.860unpack_nas_SLQSGetNetworkTime_t Struct Reference . . . . .	816



8.860.1 Detailed Description . . . . .	817
8.860.2 Field Documentation . . . . .	817
8.860.2.1 p3GPP2TimeInfo . . . . .	817
8.860.2.2 p3GPPTimeInfo . . . . .	817
8.860.2.3 ParamPresenceMask . . . . .	817
8.861unpack_nas_SLQSGetOperatorNameData_t Struct Reference . . . . .	817
8.861.1 Detailed Description . . . . .	817
8.861.2 Field Documentation . . . . .	818
8.861.2.1 ParamPresenceMask . . . . .	818
8.861.2.2 pNITZInformation . . . . .	818
8.861.2.3 pOperatorNameString . . . . .	818
8.861.2.4 pOperatorPLMNList . . . . .	818
8.861.2.5 pPLMNNetworkName . . . . .	818
8.861.2.6 pSvcProviderName . . . . .	818
8.862unpack_nas_SLQSGetPLMNName_t Struct Reference . . . . .	818
8.862.1 Detailed Description . . . . .	819
8.862.2 Field Documentation . . . . .	821
8.862.2.1 longName . . . . .	821
8.862.2.2 longNameCI . . . . .	821
8.862.2.3 longNameEn . . . . .	821
8.862.2.4 longNameLen . . . . .	821
8.862.2.5 longNameSB . . . . .	821
8.862.2.6 ParamPresenceMask . . . . .	821
8.862.2.7 shortName . . . . .	821
8.862.2.8 shortNameCI . . . . .	821
8.862.2.9 shortNameEn . . . . .	821
8.862.2.10shortNameLen . . . . .	821
8.862.2.11shortNameSB . . . . .	821
8.862.2.12spn . . . . .	821
8.862.2.13spnEncoding . . . . .	822
8.862.2.14spnLength . . . . .	822
8.863unpack_nas_SLQSGetServingSystem_t Struct Reference . . . . .	822
8.863.1 Detailed Description . . . . .	822
8.863.2 Field Documentation . . . . .	825
8.863.2.1 BasestationID . . . . .	825
8.863.2.2 BasestationLatitude . . . . .	825
8.863.2.3 BasestationLongitude . . . . .	825
8.863.2.4 CallBarStatus . . . . .	825
8.863.2.5 CDMA_P_Rev . . . . .	825
8.863.2.6 CDMASystemInfoExt . . . . .	825

8.863.2.7 CellID	826
8.863.2.8 ConcSvcInfo	826
8.863.2.9 CurrentPLMN	826
8.863.2.10DataSrvCapabilities	826
8.863.2.11DefaultRoamInd	826
8.863.2.12DetailedSvcInfo	826
8.863.2.13DTMInd	826
8.863.2.14Gpp2TimeZone	826
8.863.2.15GppNetworkDSTAdjustment	826
8.863.2.16GppTimeZone	826
8.863.2.17HdrPersonality	826
8.863.2.18Lac	826
8.863.2.19NetworkID	826
8.863.2.20ParamPresenceMask	826
8.863.2.21PRLInd	826
8.863.2.22RoamIndicatorVal	826
8.863.2.23RoamingIndicatorList	826
8.863.2.24ServingSystem	826
8.863.2.25SystemID	826
8.863.2.26TrackAreaCode	826
8.864unpack_nas_SLQSGetservingSystemV2_t Struct Reference	826
8.864.1 Detailed Description	827
8.864.2 Field Documentation	830
8.864.2.1 BasestationID	830
8.864.2.2 BasestationLatitude	830
8.864.2.3 BasestationLongitude	830
8.864.2.4 CallBarStatus	830
8.864.2.5 CDMA_P_Rev	830
8.864.2.6 CDMASystemInfoExt	830
8.864.2.7 CellID	830
8.864.2.8 ConcSvcInfo	830
8.864.2.9 CurrentPLMN	830
8.864.2.10DataSrvCapabilities	830
8.864.2.11DefaultRoamInd	830
8.864.2.12DetailedSvcInfo	830
8.864.2.13DTMInd	830
8.864.2.14Gpp2TimeZone	830
8.864.2.15GppNetworkDSTAdjustment	830
8.864.2.16GppTimeZone	830
8.864.2.17HdrPersonality	830

8.864.2.18	Lac	830
8.864.2.19	MNCPSCDigitStatTlv	830
8.864.2.20	NetworkID	831
8.864.2.21	ParamPresenceMask	831
8.864.2.22	PRLInd	831
8.864.2.23	RoamIndicatorVal	831
8.864.2.24	RoamingIndicatorList	831
8.864.2.25	ServingSystem	831
8.864.2.26	SystemID	831
8.864.2.27	TrackAreaCode	831
8.865	unpack_nas_SLQSGetSignalStrength_t Struct Reference	831
8.865.1	Detailed Description	831
8.865.2	Field Documentation	833
8.865.2.1	ecioList	833
8.865.2.2	ecioListLen	833
8.865.2.3	errorRateList	833
8.865.2.4	errorRateListLen	833
8.865.2.5	lo	833
8.865.2.6	ltsrpr	833
8.865.2.7	ltsnr	833
8.865.2.8	ParamPresenceMask	833
8.865.2.9	rsrqInfo	833
8.865.2.10	rxSignalStrengthList	834
8.865.2.11	rxSignalStrengthListLen	834
8.865.2.12	signalStrengthReqMask	834
8.865.2.13	sinr	834
8.866	unpack_nas_SLQSGetSysInfo_t Struct Reference	834
8.866.1	Detailed Description	834
8.866.2	Field Documentation	836
8.866.2.1	pAddCDMASysInfo	836
8.866.2.2	pAddGSMSysInfo	836
8.866.2.3	pAddHDRSysInfo	836
8.866.2.4	pAddLTESysInfo	837
8.866.2.5	pAddWCDMASysInfo	837
8.866.2.6	ParamPresenceMask	837
8.866.2.7	pCampedCiotLteOpMode	837
8.866.2.8	pCDMASrvStatusInfo	837
8.866.2.9	pCDMASysInfo	837
8.866.2.10	pGSMCallBarringSysInfo	837
8.866.2.11	pGSMCipherDomainSysInfo	837

8.866.2.12	pGSMsSrvStatusInfo	837
8.866.2.13	pGSMsSysInfo	837
8.866.2.14	pHDRsSrvStatusInfo	837
8.866.2.15	pHDRsSysInfo	837
8.866.2.16	pLTESrvStatusInfo	837
8.866.2.17	pLTESysInfo	837
8.866.2.18	pLTEVoiceSupportSysInfo	837
8.866.2.19	pWCDMAcCallBarringSysInfo	837
8.866.2.20	pWCDMAcCipherDomainSysInfo	837
8.866.2.21	pWCDMASrvStatusInfo	837
8.866.2.22	pWCDMASysInfo	837
8.867	unpack_nas_SLQSGetSysInfoV2_t Struct Reference	837
8.867.1	Detailed Description	838
8.867.2	Field Documentation	841
8.867.2.1	pAddCDMASysInfo	841
8.867.2.2	pAddGSMsSysInfo	841
8.867.2.3	pAddHDRsSysInfo	841
8.867.2.4	pAddLTESysInfo	841
8.867.2.5	pAddWCDMASysInfo	841
8.867.2.6	ParamPresenceMask	841
8.867.2.7	pCampedCiotLteOpMode	841
8.867.2.8	pCDMASrvStatusInfo	841
8.867.2.9	pCDMASysInfo	841
8.867.2.10	pGSMcCallBarringSysInfo	841
8.867.2.11	pGSMcCipherDomainSysInfo	841
8.867.2.12	pGSMsSrvStatusInfo	841
8.867.2.13	pGSMsSysInfo	841
8.867.2.14	pHDRsSrvStatusInfo	841
8.867.2.15	pHDRsSysInfo	841
8.867.2.16	pImmsVoiceSupportLte	841
8.867.2.17	pLteEmbmsCoverage	842
8.867.2.18	pLteEmbmsTraceId	842
8.867.2.19	pLteRegDomain	842
8.867.2.20	pLTESrvStatusInfo	842
8.867.2.21	pLTESysInfo	842
8.867.2.22	pLteVoiceDomain	842
8.867.2.23	pLTEVoiceSupportSysInfo	842
8.867.2.24	pNR5GCellStatusInfo	842
8.867.2.25	pNR5GSrvStatusInfo	842
8.867.2.26	pNR5GSysInfo	842

8.867.2.27pSimRejInfo . . . . .	842
8.867.2.28pSrvRegRestriction . . . . .	842
8.867.2.29pWCDMACallBarringSysInfo . . . . .	842
8.867.2.30pWCDMACipherDomainSysInfo . . . . .	842
8.867.2.31pWCDMASrvStatusInfo . . . . .	842
8.867.2.32pWCDMASysInfo . . . . .	842
8.868unpack_nas_SLQSGetSysSelectionPref_t Struct Reference . . . . .	842
8.868.1 Detailed Description . . . . .	842
8.868.2 Field Documentation . . . . .	847
8.868.2.1 ParamPresenceMask . . . . .	847
8.868.2.2 pBandPref . . . . .	847
8.868.2.3 pEmerMode . . . . .	847
8.868.2.4 pGWAcqOrderPref . . . . .	847
8.868.2.5 pLTEBandPref . . . . .	847
8.868.2.6 pModePref . . . . .	847
8.868.2.7 pNetSelPref . . . . .	847
8.868.2.8 pPRLPref . . . . .	847
8.868.2.9 pRoamPref . . . . .	847
8.868.2.10pSrvDomainPref . . . . .	847
8.869unpack_nas_SLQSGetSysSelectionPrefExt_t Struct Reference . . . . .	847
8.869.1 Detailed Description . . . . .	848
8.869.2 Field Documentation . . . . .	849
8.869.2.1 pAcqOrderPref . . . . .	849
8.869.2.2 pBandPref . . . . .	849
8.869.2.3 pCiotAcqOrderPref . . . . .	849
8.869.2.4 pCiotLteOpModePref . . . . .	850
8.869.2.5 pEmerMode . . . . .	850
8.869.2.6 pGWAcqOrderPref . . . . .	850
8.869.2.7 pLTEBandPref . . . . .	850
8.869.2.8 pLTEBandPrefExt . . . . .	850
8.869.2.9 pLteM1BandPref . . . . .	850
8.869.2.10pLteNb1BandPref . . . . .	850
8.869.2.11pModePref . . . . .	850
8.869.2.12pNetSelPref . . . . .	850
8.869.2.13pNr5gBandPref . . . . .	850
8.869.2.14pPRLPref . . . . .	850
8.869.2.15pRatDisabledMask . . . . .	850
8.869.2.16pRoamPref . . . . .	850
8.869.2.17pSrvDomainPref . . . . .	850
8.870unpack_nas_SLQSGetSysSelectionPrefExtV2_t Struct Reference . . . . .	850

8.870.1 Detailed Description . . . . .	851
8.870.2 Field Documentation . . . . .	853
8.870.2.1 pAcqOrderPref . . . . .	853
8.870.2.2 ParamPresenceMask . . . . .	853
8.870.2.3 pBandPref . . . . .	853
8.870.2.4 pCiotAcqOrderPref . . . . .	853
8.870.2.5 pCiotLteOpModePref . . . . .	853
8.870.2.6 pEmerMode . . . . .	853
8.870.2.7 pGWAcqOrderPref . . . . .	853
8.870.2.8 pLTEBandPref . . . . .	853
8.870.2.9 pLTEBandPrefExt . . . . .	853
8.870.2.10pLteM1BandPref . . . . .	853
8.870.2.11pLteNb1BandPref . . . . .	853
8.870.2.12pModePref . . . . .	853
8.870.2.13pNetSelPref . . . . .	853
8.870.2.14pNr5gBandPref . . . . .	853
8.870.2.15pPRLPref . . . . .	853
8.870.2.16pRatDisabledMask . . . . .	853
8.870.2.17pRoamPref . . . . .	853
8.870.2.18pSrvDomainPref . . . . .	853
8.870.2.19pSrvRegRestric . . . . .	853
8.870.2.20pTDSCDMABandPref . . . . .	854
8.870.2.21pUsageSetting . . . . .	854
8.870.2.22pVoiceDomainPref . . . . .	854
8.871unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t Struct Reference . . . . .	854
8.871.1 Detailed Description . . . . .	854
8.871.2 Field Documentation . . . . .	854
8.871.2.1 edrxCiotLteModeTlv . . . . .	854
8.871.2.2 edrxCycleLengthTlv . . . . .	854
8.871.2.3 edrxEnableTypeTlv . . . . .	855
8.871.2.4 edrxPagingTimeWindowTlv . . . . .	855
8.871.2.5 edrxRatTypeTlv . . . . .	855
8.871.2.6 ParamPresenceMask . . . . .	855
8.872unpack_nas_SLQSNasGet3GPP2Subscription_t Struct Reference . . . . .	855
8.872.1 Detailed Description . . . . .	855
8.872.2 Field Documentation . . . . .	856
8.872.2.1 ParamPresenceMask . . . . .	856
8.872.2.2 pCDMAChannel . . . . .	856
8.872.2.3 pDirNum . . . . .	856
8.872.2.4 pHomeSIDNID . . . . .	856

8.872.2.5 pMdn . . . . .	856
8.872.2.6 pMinBasedIMSI . . . . .	856
8.872.2.7 pNAMNameInfo . . . . .	856
8.872.2.8 pTrueIMSI . . . . .	856
8.873unpack_nas_SLQSNasGetCellLocationInfo_t Struct Reference . . . . .	856
8.873.1 Detailed Description . . . . .	856
8.873.2 Field Documentation . . . . .	857
8.873.2.1 ParamPresenceMask . . . . .	857
8.873.2.2 pCDMAInfo . . . . .	857
8.873.2.3 pGERANInfo . . . . .	857
8.873.2.4 pLTEInfoInterfreq . . . . .	857
8.873.2.5 pLTEInfoIntrafreq . . . . .	857
8.873.2.6 pLTEInfoNeighboringGSM . . . . .	857
8.873.2.7 pLTEInfoNeighboringWCDMA . . . . .	857
8.873.2.8 pUMTSCellID . . . . .	858
8.873.2.9 pUMTSInfo . . . . .	858
8.873.2.10pWCDMAInfoLTENeighborCell . . . . .	858
8.874unpack_nas_SLQSNasGetCellLocationInfoV2_t Struct Reference . . . . .	858
8.874.1 Detailed Description . . . . .	858
8.874.2 Field Documentation . . . . .	859
8.874.2.1 ParamPresenceMask . . . . .	859
8.874.2.2 pCDMAInfo . . . . .	859
8.874.2.3 pGERANInfo . . . . .	859
8.874.2.4 pLteEarfcnInfo . . . . .	859
8.874.2.5 pLTEInfoInterfreq . . . . .	859
8.874.2.6 pLTEInfoIntrafreq . . . . .	859
8.874.2.7 pLTEInfoNeighboringGSM . . . . .	859
8.874.2.8 pLTEInfoNeighboringWCDMA . . . . .	859
8.874.2.9 pUMTSCellID . . . . .	859
8.874.2.10pUMTSExtInfo . . . . .	859
8.874.2.11pUMTSInfo . . . . .	860
8.874.2.12pWCDMACellInfoExt . . . . .	860
8.874.2.13pWCDMAInfoLTENeighborCell . . . . .	860
8.875unpack_nas_SLQSNASGeteDRXParams_t Struct Reference . . . . .	860
8.875.1 Detailed Description . . . . .	860
8.875.2 Field Documentation . . . . .	860
8.875.2.1 ParamPresenceMask . . . . .	860
8.875.2.2 pCycleLen . . . . .	860
8.875.2.3 pEdrxEnable . . . . .	860
8.875.2.4 pPagingTimeWindow . . . . .	860

8.876unpack_nas_SLQSNASGeteDRXParamsExt_t Struct Reference . . . . .	861
8.876.1 Detailed Description . . . . .	861
8.876.2 Field Documentation . . . . .	862
8.876.2.1 ParamPresenceMask . . . . .	862
8.876.2.2 pCycleLen . . . . .	862
8.876.2.3 pEdrxEnable . . . . .	862
8.876.2.4 pEdrxRAT . . . . .	862
8.876.2.5 pLteOpMode . . . . .	862
8.876.2.6 pPagingTimeWindow . . . . .	862
8.877unpack_nas_SLQSNASGetForbiddenNetworks_t Struct Reference . . . . .	862
8.877.1 Detailed Description . . . . .	862
8.877.2 Field Documentation . . . . .	862
8.877.2.1 ParamPresenceMask . . . . .	862
8.877.2.2 pForbiddenNetworks3GPP . . . . .	863
8.878unpack_nas_SLQSNasGetHDRColorCode_t Struct Reference . . . . .	863
8.878.1 Detailed Description . . . . .	863
8.878.2 Field Documentation . . . . .	863
8.878.2.1 ParamPresenceMask . . . . .	863
8.878.2.2 pColorCode . . . . .	863
8.879unpack_nas_SLQSNasGetRFInfo_t Struct Reference . . . . .	863
8.879.1 Detailed Description . . . . .	863
8.879.2 Field Documentation . . . . .	864
8.879.2.1 pLTEOperationMode . . . . .	864
8.879.2.2 pRfBandInfoExtFormat . . . . .	864
8.879.2.3 pRfBandwidthInfo . . . . .	864
8.879.2.4 pRfDedicatedBandInfo . . . . .	864
8.879.2.5 rfbandInfoList . . . . .	864
8.879.2.6 Tlvresult . . . . .	864
8.880unpack_nas_SLQSNasGetSigInfo_t Struct Reference . . . . .	864
8.880.1 Detailed Description . . . . .	864
8.880.2 Field Documentation . . . . .	865
8.880.2.1 CDMASSTInfo . . . . .	865
8.880.2.2 GSMSSInfo . . . . .	865
8.880.2.3 HDRSSInfo . . . . .	865
8.880.2.4 LTESSTInfo . . . . .	865
8.880.2.5 ParamPresenceMask . . . . .	865
8.880.2.6 WCDMASSInfo . . . . .	865
8.881unpack_nas_SLQSNasGetTxRxInfo_t Struct Reference . . . . .	865
8.881.1 Detailed Description . . . . .	865
8.881.2 Field Documentation . . . . .	866



8.881.2.1 ParamPresenceMask . . . . .	866
8.881.2.2 pRXChain0Info . . . . .	866
8.881.2.3 pRXChain1Info . . . . .	866
8.881.2.4 pTXInfo . . . . .	866
8.882unpack_nas_SLQSNasNetworkRejectCallback_Ind_t Struct Reference . . . . .	866
8.882.1 Detailed Description . . . . .	866
8.882.2 Field Documentation . . . . .	869
8.882.2.1 ParamPresenceMask . . . . .	869
8.882.2.2 pCsgId . . . . .	869
8.882.2.3 pLteOpMode . . . . .	869
8.882.2.4 pPlmnId . . . . .	869
8.882.2.5 radiolfl . . . . .	869
8.882.2.6 rejectCause . . . . .	869
8.882.2.7 serviceDomain . . . . .	869
8.883unpack_nas_SLQSNasNetworkTimeCallBack_ind_t Struct Reference . . . . .	869
8.883.1 Detailed Description . . . . .	869
8.883.2 Field Documentation . . . . .	870
8.883.2.1 ParamPresenceMask . . . . .	870
8.883.2.2 pDayltSavAdj . . . . .	870
8.883.2.3 pRadioInterface . . . . .	870
8.883.2.4 pTimeZone . . . . .	870
8.883.2.5 universalTime . . . . .	870
8.884unpack_nas_SLQSNasRFBandInfoCallback_Ind_t Struct Reference . . . . .	870
8.884.1 Detailed Description . . . . .	871
8.884.2 Field Documentation . . . . .	871
8.884.2.1 lteOperationalMode . . . . .	871
8.884.2.2 ParamPresenceMask . . . . .	871
8.884.2.3 rfBandInfo . . . . .	871
8.884.2.4 rfBandInfoExt . . . . .	871
8.884.2.5 rfBandwidthInfo . . . . .	871
8.884.2.6 rfDedicatedBandInfo . . . . .	871
8.885unpack_nas_SLQSNasSigInfoCallback_ind_t Struct Reference . . . . .	871
8.885.1 Detailed Description . . . . .	872
8.885.2 Field Documentation . . . . .	872
8.885.2.1 ParamPresenceMask . . . . .	872
8.885.2.2 pCDMASigInfo . . . . .	872
8.885.2.3 pGSMSigInfo . . . . .	873
8.885.2.4 pHDRSigInfo . . . . .	873
8.885.2.5 pLTESigInfo . . . . .	873
8.885.2.6 pRscp . . . . .	873

8.885.2.7 pTDSCDMASigInfoExt . . . . .	873
8.885.2.8 pWCDMASigInfo . . . . .	873
8.886unpack_nas_SLQSNASSwiGetChannelLock_t Struct Reference . . . . .	873
8.886.1 Detailed Description . . . . .	873
8.886.2 Field Documentation . . . . .	873
8.886.2.1 ParamPresenceMask . . . . .	873
8.886.2.2 pLteEARFCN . . . . .	873
8.886.2.3 pLtePCI . . . . .	873
8.886.2.4 pWcdmaUARFCN . . . . .	873
8.887unpack_nas_SLQSNasSwiModemStatus_t Struct Reference . . . . .	874
8.887.1 Detailed Description . . . . .	874
8.887.2 Field Documentation . . . . .	874
8.887.2.1 commonInfo . . . . .	874
8.887.2.2 ParamPresenceMask . . . . .	874
8.887.2.3 pLTEInfo . . . . .	874
8.888unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t Struct Reference . . . . .	874
8.888.1 Detailed Description . . . . .	874
8.888.2 Field Documentation . . . . .	875
8.888.2.1 Info . . . . .	875
8.888.2.2 ParamPresenceMask . . . . .	875
8.888.2.3 Tlvresult . . . . .	875
8.889unpack_nas_SLQSNasTimerCallback_ind_t Struct Reference . . . . .	875
8.889.1 Detailed Description . . . . .	875
8.889.2 Field Documentation . . . . .	875
8.889.2.1 ParamPresenceMask . . . . .	875
8.889.2.2 t3396_apn . . . . .	876
8.889.2.3 t3396_plmn_id . . . . .	876
8.889.2.4 t3396_val . . . . .	876
8.890unpack_nas_SLQSPerformNetworkScanV2_t Struct Reference . . . . .	876
8.890.1 Detailed Description . . . . .	876
8.890.2 Field Documentation . . . . .	877
8.890.2.1 p3GppNetworkInfoInstances . . . . .	877
8.890.2.2 p3GppNetworkInstanceSize . . . . .	877
8.890.2.3 ParamPresenceMask . . . . .	877
8.890.2.4 pLteOpModeTlv . . . . .	877
8.890.2.5 pNetworkNameSrcTlv . . . . .	877
8.890.2.6 pPCIInfo . . . . .	877
8.890.2.7 pPCSInstance . . . . .	877
8.890.2.8 pPCSInstanceSize . . . . .	877
8.890.2.9 pRATINstance . . . . .	877

8.890.2.1pRATInstanceSize . . . . .	877
8.890.2.1pScanResult . . . . .	878
8.891unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t Struct Reference . . . . .	878
8.891.1 Detailed Description . . . . .	878
8.891.2 Field Documentation . . . . .	878
8.891.2.1 Info . . . . .	878
8.891.2.2 ParamPresenceMask . . . . .	878
8.891.2.3 Tlvresult . . . . .	878
8.892unpack_nas_SLQSSwiGetHDRPersonality_t Struct Reference . . . . .	878
8.892.1 Detailed Description . . . . .	878
8.892.2 Field Documentation . . . . .	879
8.892.2.1 ParamPresenceMask . . . . .	879
8.892.2.2 pCurrentPersonality . . . . .	879
8.892.2.3 pPersonalityListLength . . . . .	879
8.892.2.4 pProtocolSubtypeElement . . . . .	879
8.893unpack_nas_SLQSSwiGetHDRProtSubtype_t Struct Reference . . . . .	879
8.893.1 Detailed Description . . . . .	879
8.893.2 Field Documentation . . . . .	880
8.893.2.1 pAppSubType . . . . .	880
8.893.2.2 ParamPresenceMask . . . . .	880
8.893.2.3 pCurrentPrsnlty . . . . .	880
8.893.2.4 pPersonalityListLength . . . . .	880
8.893.2.5 pProtoSubTypElmnt . . . . .	880
8.894unpack_nas_SLQSSwiGetHRPDStats_t Struct Reference . . . . .	880
8.894.1 Detailed Description . . . . .	880
8.894.2 Field Documentation . . . . .	881
8.894.2.1 ParamPresenceMask . . . . .	881
8.894.2.2 pDRCPParams . . . . .	881
8.894.2.3 pPilotSetData . . . . .	881
8.894.2.4 pUATI . . . . .	881
8.895unpack_nas_SLQSSwiGetLteCQI_t Struct Reference . . . . .	881
8.895.1 Detailed Description . . . . .	881
8.895.2 Field Documentation . . . . .	882
8.895.2.1 CQIValueCW0 . . . . .	882
8.895.2.2 CQIValueCW1 . . . . .	882
8.895.2.3 ParamPresenceMask . . . . .	882
8.895.2.4 ValidityCW0 . . . . .	882
8.895.2.5 ValidityCW1 . . . . .	882
8.896unpack_nas_SLQSSwiGetLteScCRxInfo_t Struct Reference . . . . .	882
8.896.1 Detailed Description . . . . .	882

8.896.2 Field Documentation . . . . .	882
8.896.2.1 ParamPresenceMask . . . . .	882
8.896.2.2 pScCRxInfo . . . . .	882
8.897unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t Struct Reference . . . . .	882
8.897.1 Detailed Description . . . . .	883
8.897.2 Field Documentation . . . . .	883
8.897.2.1 ParamPresenceMask . . . . .	883
8.897.2.2 pHDRPersInd . . . . .	883
8.898unpack_nas_SLQSSwiNetworkDebug_t Struct Reference . . . . .	883
8.898.1 Detailed Description . . . . .	883
8.898.2 Field Documentation . . . . .	884
8.898.2.1 ParamPresenceMask . . . . .	884
8.898.2.2 pDataStatusDetail . . . . .	884
8.898.2.3 pDeviceConfigDetail . . . . .	884
8.898.2.4 pNetworkStat1x . . . . .	884
8.898.2.5 pNetworkStatEVDO . . . . .	884
8.898.2.6 pObjectVer . . . . .	884
8.899unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t Struct Reference . . . . .	884
8.899.1 Detailed Description . . . . .	884
8.899.2 Field Documentation . . . . .	884
8.899.2.1 ParamPresenceMask . . . . .	884
8.899.2.2 rankIndicatorTlv . . . . .	884
8.900unpack_nas_SLQSSysInfoCallback_ind_t Struct Reference . . . . .	884
8.900.1 Detailed Description . . . . .	885
8.900.2 Field Documentation . . . . .	887
8.900.2.1 pAddCDMASysInfo . . . . .	888
8.900.2.2 pAddGSMSysInfo . . . . .	888
8.900.2.3 pAddHDRSysInfo . . . . .	888
8.900.2.4 pAddLTESysInfo . . . . .	888
8.900.2.5 pAddWCDMASysInfo . . . . .	888
8.900.2.6 ParamPresenceMask . . . . .	888
8.900.2.7 pCDMASrvStatusInfo . . . . .	888
8.900.2.8 pCDMASysInfo . . . . .	888
8.900.2.9 pGSMCallBarringSysInfo . . . . .	888
8.900.2.10pGSMCipherDomainSysInfo . . . . .	888
8.900.2.11pGSMSrvStatusInfo . . . . .	888
8.900.2.12pGSMSysInfo . . . . .	888
8.900.2.13pHDRSrvStatusInfo . . . . .	888
8.900.2.14pHDRSysInfo . . . . .	888
8.900.2.15pLteCiotOpModeTlv . . . . .	888

8.900.2.16	lteSrvStatusInfo	888
8.900.2.17	lteSysInfo	888
8.900.2.18	lteVoiceSupportSysInfo	888
8.900.2.19	nr5GCellStatus	888
8.900.2.20	nr5GSerStatTlv	888
8.900.2.21	nr5GSystemInfoTlv	888
8.900.2.22	sysInfoNoChange	888
8.900.2.23	wcdmaCallBarringSysInfo	888
8.900.2.24	wcdmaCipherDomainSysInfo	888
8.900.2.25	wcdmaSrvStatusInfo	888
8.900.2.26	wcdmaSysInfo	888
8.901	unpack_omaDmConfigTlv_t Struct Reference	888
8.901.1	Detailed Description	889
8.901.2	Field Documentation	889
8.901.2.1	alertmsg	889
8.901.2.2	alertmsglength	889
8.901.2.3	state	889
8.901.2.4	userInputReq	889
8.901.2.5	userInputTimeout	889
8.902	unpack_omaDmFotaTlv_t Struct Reference	889
8.902.1	Detailed Description	890
8.902.2	Field Documentation	891
8.902.2.1	description	891
8.902.2.2	descriptionlength	891
8.902.2.3	fwdloadsize	891
8.902.2.4	fwloadComplete	891
8.902.2.5	namelength	891
8.902.2.6	package_name	891
8.902.2.7	sessionType	891
8.902.2.8	severity	891
8.902.2.9	state	891
8.902.2.10	updateCompleteStatus	891
8.902.2.11	userInputReq	891
8.902.2.12	userInputTimeout	891
8.902.2.13	version	891
8.902.2.14	versionlength	891
8.903	unpack_omaDmNotificationsTlv_t Struct Reference	891
8.903.1	Detailed Description	892
8.903.2	Field Documentation	892
8.903.2.1	notification	892

8.903.2.2 sessionStatus . . . . .	892
8.904unpack_pds_GetPDSDDefaults_t Struct Reference . . . . .	892
8.904.1 Detailed Description . . . . .	892
8.904.2 Field Documentation . . . . .	893
8.904.2.1 pAccuracy . . . . .	893
8.904.2.2 ParamPresenceMask . . . . .	893
8.904.2.3 plInterval . . . . .	893
8.904.2.4 pOperation . . . . .	893
8.904.2.5 pTimeout . . . . .	893
8.905unpack_pds_GetPDSSState_t Struct Reference . . . . .	893
8.905.1 Detailed Description . . . . .	893
8.905.2 Field Documentation . . . . .	894
8.905.2.1 ParamPresenceMask . . . . .	894
8.905.2.2 pEnabledStatus . . . . .	894
8.905.2.3 pTrackingStatus . . . . .	894
8.906unpack_pds_GetPortAutomaticTracking_t Struct Reference . . . . .	894
8.906.1 Detailed Description . . . . .	894
8.906.2 Field Documentation . . . . .	894
8.906.2.1 ParamPresenceMask . . . . .	895
8.906.2.2 pbAuto . . . . .	895
8.907unpack_pds_GetServiceAutomaticTracking_t Struct Reference . . . . .	895
8.907.1 Detailed Description . . . . .	895
8.907.2 Field Documentation . . . . .	895
8.907.2.1 ParamPresenceMask . . . . .	895
8.907.2.2 pbAuto . . . . .	895
8.908unpack_pds_GetXTRAAutomaticDownload_t Struct Reference . . . . .	895
8.908.1 Detailed Description . . . . .	895
8.908.2 Field Documentation . . . . .	896
8.908.2.1 ParamPresenceMask . . . . .	896
8.908.2.2 pbEnabled . . . . .	896
8.908.2.3 plInterval . . . . .	896
8.909unpack_pds_GetXTRANetwork_t Struct Reference . . . . .	896
8.909.1 Detailed Description . . . . .	896
8.909.2 Field Documentation . . . . .	896
8.909.2.1 ParamPresenceMask . . . . .	896
8.909.2.2 pPreference . . . . .	896
8.910unpack_pds_GetXTRAVality_t Struct Reference . . . . .	896
8.910.1 Detailed Description . . . . .	897
8.910.2 Field Documentation . . . . .	897
8.910.2.1 ParamPresenceMask . . . . .	897

8.910.2.2 pDuration . . . . .	897
8.910.2.3 pGPSWeek . . . . .	897
8.910.2.4 pGPSWeekOffset . . . . .	897
8.911unpack_pds_SetEventReport_Ind_t Struct Reference . . . . .	897
8.911.1 Detailed Description . . . . .	898
8.911.2 Field Documentation . . . . .	899
8.911.2.1 dLatitude . . . . .	899
8.911.2.2 dLongitude . . . . .	899
8.911.2.3 has_dLatitude . . . . .	899
8.911.2.4 has_dLongitude . . . . .	899
8.911.2.5 has_PositionDataNMEA . . . . .	899
8.911.2.6 has_posSrc . . . . .	899
8.911.2.7 has_SessionStatus . . . . .	899
8.911.2.8 ParamPresenceMask . . . . .	899
8.911.2.9 PositionDataNMEA . . . . .	899
8.911.2.10posSrc . . . . .	899
8.911.2.11SessionStatus . . . . .	899
8.912unpack_pds_SetPdsState_Ind_t Struct Reference . . . . .	899
8.912.1 Detailed Description . . . . .	899
8.912.2 Field Documentation . . . . .	900
8.912.2.1 EnabledStatus . . . . .	900
8.912.2.2 ParamPresenceMask . . . . .	900
8.912.2.3 TlvPresent . . . . .	900
8.912.2.4 TrackingStatus . . . . .	900
8.913unpack_pds_SLQSGetAGPSConfig_t Struct Reference . . . . .	900
8.913.1 Detailed Description . . . . .	900
8.913.2 Field Documentation . . . . .	901
8.913.2.1 ParamPresenceMask . . . . .	901
8.913.2.2 pServerAddress . . . . .	901
8.913.2.3 pServerPort . . . . .	901
8.913.2.4 pServerURL . . . . .	901
8.913.2.5 pServerURLLength . . . . .	901
8.914unpack_pds_SLQSGetGPSSStateInfo_t Struct Reference . . . . .	901
8.914.1 Detailed Description . . . . .	902
8.914.2 Field Documentation . . . . .	905
8.914.2.1 Altitude . . . . .	905
8.914.2.2 EngineState . . . . .	905
8.914.2.3 glo_almanac_sv_msk . . . . .	905
8.914.2.4 glo_ephemeris_sv_msk . . . . .	905
8.914.2.5 glo_health_sv_msk . . . . .	905

8.914.2.6 glo_visible_sv_msk . . . . .	905
8.914.2.7 gps_almanac_sv_msk . . . . .	905
8.914.2.8 gps_ephemeris_sv_msk . . . . .	905
8.914.2.9 gps_health_sv_msk . . . . .	905
8.914.2.10gps_visible_sv_msk . . . . .	905
8.914.2.11HorizontalUncertainty . . . . .	905
8.914.2.12ono_valid . . . . .	905
8.914.2.13Latitude . . . . .	905
8.914.2.14Longitude . . . . .	905
8.914.2.15ParamPresenceMask . . . . .	905
8.914.2.16sbas_almanac_sv_msk . . . . .	905
8.914.2.17sbas_ephemeris_sv_msk . . . . .	905
8.914.2.18sbas_health_sv_msk . . . . .	905
8.914.2.19sbas_visible_sv_msk . . . . .	905
8.914.2.20Time_uncert_ms . . . . .	905
8.914.2.21TimeStmp_gps_week . . . . .	905
8.914.2.22TimeStmp_tow_ms . . . . .	905
8.914.2.23ValidMask . . . . .	905
8.914.2.24VerticalUncertainty . . . . .	905
8.914.2.25xtra_start_gps_minutes . . . . .	906
8.914.2.26xtra_start_gps_week . . . . .	906
8.914.2.27xtra_valid_duration_hours . . . . .	906
8.915unpack_qmi_t Struct Reference . . . . .	906
8.915.1 Detailed Description . . . . .	906
8.915.2 Field Documentation . . . . .	906
8.915.2.1 msgid . . . . .	906
8.915.2.2 type . . . . .	906
8.915.2.3 xid . . . . .	906
8.916unpack_qos_BindDataPort_t Struct Reference . . . . .	906
8.916.1 Detailed Description . . . . .	906
8.916.2 Field Documentation . . . . .	906
8.916.2.1 ParamPresenceMask . . . . .	906
8.917unpack_qos_dataRate_t Struct Reference . . . . .	907
8.917.1 Detailed Description . . . . .	907
8.917.2 Field Documentation . . . . .	907
8.917.2.1 dataRateMax . . . . .	907
8.917.2.2 guaranteedRate . . . . .	907
8.918unpack_qos_IPv4Addr_t Struct Reference . . . . .	907
8.918.1 Detailed Description . . . . .	907
8.918.2 Field Documentation . . . . .	907



8.918.2.1 addr . . . . .	907
8.918.2.2 subnetMask . . . . .	907
8.919unpack_qos_IPv6Addr_t Struct Reference . . . . .	907
8.919.1 Detailed Description . . . . .	908
8.919.2 Field Documentation . . . . .	908
8.919.2.1 addr . . . . .	908
8.919.2.2 prefixLen . . . . .	908
8.920unpack_qos_IPv6TrafCls_t Struct Reference . . . . .	908
8.920.1 Detailed Description . . . . .	908
8.920.2 Field Documentation . . . . .	908
8.920.2.1 mask . . . . .	908
8.920.2.2 val . . . . .	908
8.921unpack_qos_pktErrRate_t Struct Reference . . . . .	908
8.921.1 Detailed Description . . . . .	909
8.921.2 Field Documentation . . . . .	909
8.921.2.1 exponent . . . . .	909
8.921.2.2 multiplier . . . . .	909
8.922unpack_qos_Port_t Struct Reference . . . . .	909
8.922.1 Detailed Description . . . . .	909
8.922.2 Field Documentation . . . . .	909
8.922.2.1 port . . . . .	909
8.922.2.2 range . . . . .	909
8.923unpack_qos_QosFlowInfo_t Struct Reference . . . . .	909
8.923.1 Detailed Description . . . . .	910
8.923.2 Field Documentation . . . . .	910
8.923.2.1 BearerID . . . . .	910
8.923.2.2 is_RxQFlowGranted_Available . . . . .	910
8.923.2.3 is_TxQFlowGranted_Available . . . . .	910
8.923.2.4 NumRxFilters . . . . .	911
8.923.2.5 NumTxFilters . . . . .	911
8.923.2.6 QFlowState . . . . .	911
8.923.2.7 RxQFilter . . . . .	911
8.923.2.8 RxQFlowGranted . . . . .	911
8.923.2.9 TxQFilter . . . . .	911
8.923.2.10TxQFlowGranted . . . . .	911
8.924unpack_qos_QosFlowInfoState_t Struct Reference . . . . .	911
8.924.1 Detailed Description . . . . .	911
8.924.2 Field Documentation . . . . .	911
8.924.2.1 id . . . . .	911
8.924.2.2 isNewFlow . . . . .	911

8.924.2.3 state . . . . .	911
8.925unpack_qos_SLQSQosGetNetworkStatus_t Struct Reference . . . . .	911
8.925.1 Detailed Description . . . . .	912
8.925.2 Field Documentation . . . . .	912
8.925.2.1 NWQoSStatus . . . . .	912
8.925.2.2 ParamPresenceMask . . . . .	912
8.926unpack_qos_SLQSQosSwiReadApnExtraParams_t Struct Reference . . . . .	912
8.926.1 Detailed Description . . . . .	912
8.926.2 Field Documentation . . . . .	913
8.926.2.1 ambr_dl . . . . .	913
8.926.2.2 ambr_dl_ext . . . . .	913
8.926.2.3 ambr_dl_ext2 . . . . .	913
8.926.2.4 ambr_ul . . . . .	913
8.926.2.5 ambr_ul_ext . . . . .	913
8.926.2.6 ambr_ul_ext2 . . . . .	913
8.926.2.7 apnId . . . . .	913
8.926.2.8 ParamPresenceMask . . . . .	913
8.927unpack_qos_SLQSQosSwiReadDataStats_t Struct Reference . . . . .	914
8.927.1 Detailed Description . . . . .	914
8.927.2 Field Documentation . . . . .	915
8.927.2.1 apnId . . . . .	915
8.927.2.2 numQosFlow . . . . .	915
8.927.2.3 ParamPresenceMask . . . . .	915
8.927.2.4 qosFlow . . . . .	915
8.927.2.5 total_rx_bytes . . . . .	915
8.927.2.6 total_rx_pkt . . . . .	915
8.927.2.7 total_tx_bytes . . . . .	915
8.927.2.8 total_tx_bytes_drp . . . . .	915
8.927.2.9 total_tx_pkt . . . . .	915
8.927.2.10total_tx_pkt_drp . . . . .	915
8.928unpack_qos_SLQSSetQosEventCallback_ind_t Struct Reference . . . . .	915
8.928.1 Detailed Description . . . . .	915
8.928.2 Field Documentation . . . . .	916
8.928.2.1 NumFlows . . . . .	916
8.928.2.2 ParamPresenceMask . . . . .	916
8.928.2.3 QosFlowInfo . . . . .	916
8.929unpack_qos_SLQSSetQosEventCallback_t Struct Reference . . . . .	916
8.929.1 Detailed Description . . . . .	916
8.929.2 Field Documentation . . . . .	916
8.929.2.1 ParamPresenceMask . . . . .	916

8.930unpack_qos_SLQSSetQosNWStatusCallback_ind_t Struct Reference . . . . .	916
8.930.1 Detailed Description . . . . .	916
8.930.2 Field Documentation . . . . .	917
8.930.2.1 ParamPresenceMask . . . . .	917
8.930.2.2 status . . . . .	917
8.931unpack_qos_SLQSSetQosPriEventCallback_ind_t Struct Reference . . . . .	917
8.931.1 Detailed Description . . . . .	917
8.931.2 Field Documentation . . . . .	917
8.931.2.1 event . . . . .	917
8.931.2.2 ParamPresenceMask . . . . .	917
8.932unpack_qos_SLQSSetQosStatusCallback_ind_t Struct Reference . . . . .	917
8.932.1 Detailed Description . . . . .	918
8.932.2 Field Documentation . . . . .	919
8.932.2.1 event . . . . .	919
8.932.2.2 id . . . . .	919
8.932.2.3 ParamPresenceMask . . . . .	919
8.932.2.4 reason . . . . .	919
8.932.2.5 status . . . . .	919
8.933unpack_qos_swiQosFilter_t Struct Reference . . . . .	919
8.933.1 Detailed Description . . . . .	920
8.933.2 Field Documentation . . . . .	921
8.933.2.1 EspSpi . . . . .	921
8.933.2.2 Id . . . . .	921
8.933.2.3 index . . . . .	921
8.933.2.4 IPv4DstAddr . . . . .	921
8.933.2.5 IPv4SrcAddr . . . . .	921
8.933.2.6 IPv4Tos . . . . .	921
8.933.2.7 IPv6DstAddr . . . . .	921
8.933.2.8 IPv6Label . . . . .	921
8.933.2.9 IPv6SrcAddr . . . . .	921
8.933.2.10IPv6TrafCls . . . . .	921
8.933.2.11is_EspSpi_Available . . . . .	921
8.933.2.12s_Id_Available . . . . .	921
8.933.2.13s_IPv4DstAddr_Available . . . . .	921
8.933.2.14s_IPv4SrcAddr_Available . . . . .	921
8.933.2.15s_IPv4Tos_Available . . . . .	921
8.933.2.16s_IPv6DstAddr_Available . . . . .	921
8.933.2.17s_IPv6Label_Available . . . . .	921
8.933.2.18s_IPv6SrcAddr_Available . . . . .	921
8.933.2.19s_IPv6TrafCls_Available . . . . .	922

8.933.2.20s_NxtHdrProto_Available . . . . .	922
8.933.2.21is_Precedence_Available . . . . .	922
8.933.2.22s_TCPDstPort_Available . . . . .	922
8.933.2.23s_TCPSrcPort_Available . . . . .	922
8.933.2.24s_TranDstPort_Available . . . . .	922
8.933.2.25s_TranSrcPort_Available . . . . .	922
8.933.2.26s_UDPDstPort_Available . . . . .	922
8.933.2.27s_UDPSrcPort_Available . . . . .	922
8.933.2.28NxtHdrProto . . . . .	922
8.933.2.29Precedence . . . . .	922
8.933.2.30TCPDstPort . . . . .	922
8.933.2.31TCPSrcPort . . . . .	922
8.933.2.32TranDstPort . . . . .	922
8.933.2.33TranSrcPort . . . . .	922
8.933.2.34UDPDstPort . . . . .	922
8.933.2.35UDPSrcPort . . . . .	922
8.933.2.36version . . . . .	922
8.934unpack_qos_swiQosFlow_t Struct Reference . . . . .	922
8.934.1 Detailed Description . . . . .	923
8.934.2 Field Documentation . . . . .	925
8.934.2.1 DataRate . . . . .	925
8.934.2.2 index . . . . .	925
8.934.2.3 is_DataRate_Available . . . . .	925
8.934.2.4 is_Jitter_Available . . . . .	925
8.934.2.5 is_Latency_Available . . . . .	925
8.934.2.6 is_LteQci_Available . . . . .	925
8.934.2.7 is_MaxAllowedPktSz_Available . . . . .	925
8.934.2.8 is_MinPolicedPktSz_Available . . . . .	925
8.934.2.9 is_PktErrRate_Available . . . . .	925
8.934.2.10s_ProfileId3GPP2_Available . . . . .	925
8.934.2.11is_TokenBucket_Available . . . . .	925
8.934.2.12s_TrafficClass_Available . . . . .	925
8.934.2.13s_val_3GPP2Pri_Available . . . . .	925
8.934.2.14s_val_3GPPImCn_Available . . . . .	925
8.934.2.15s_val_3GPPResResidualBER_Available . . . . .	926
8.934.2.16s_val_3GPPSigInd_Available . . . . .	926
8.934.2.17s_val_3GPPTraHdlPri_Available . . . . .	926
8.934.2.18Jitter . . . . .	926
8.934.2.19Latency . . . . .	926
8.934.2.20LteQci . . . . .	926

8.934.2.21MaxAllowedPktSz . . . . .	926
8.934.2.22MinPolicedPktSz . . . . .	926
8.934.2.23PktErrRate . . . . .	926
8.934.2.24ProfileId3GPP2 . . . . .	926
8.934.2.25TokenBucket . . . . .	926
8.934.2.26TrafficClass . . . . .	926
8.934.2.27val_3GPP2Pri . . . . .	926
8.934.2.28val_3GPPImCn . . . . .	926
8.934.2.29val_3GPPResResidualBER . . . . .	926
8.934.2.30val_3GPPSigInd . . . . .	926
8.934.2.31val_3GPPTraHdlPri . . . . .	926
8.935unpack_qos_tokenBucket_t Struct Reference . . . . .	926
8.935.1 Detailed Description . . . . .	926
8.935.2 Field Documentation . . . . .	927
8.935.2.1 bucketSz . . . . .	927
8.935.2.2 peakRate . . . . .	927
8.935.2.3 tokenRate . . . . .	927
8.936unpack_qos_Tos_t Struct Reference . . . . .	927
8.936.1 Detailed Description . . . . .	927
8.936.2 Field Documentation . . . . .	927
8.936.2.1 mask . . . . .	927
8.936.2.2 val . . . . .	927
8.937unpack_QosFlowStat_t Struct Reference . . . . .	927
8.937.1 Detailed Description . . . . .	927
8.937.2 Field Documentation . . . . .	928
8.937.2.1 bearerId . . . . .	928
8.937.2.2 tx_bytes . . . . .	928
8.937.2.3 tx_bytes_drp . . . . .	928
8.937.2.4 tx_pkt . . . . .	928
8.937.2.5 tx_pkt_drp . . . . .	928
8.938unpack_result_t Struct Reference . . . . .	928
8.938.1 Detailed Description . . . . .	928
8.938.2 Field Documentation . . . . .	928
8.938.2.1 ParamPresenceMask . . . . .	928
8.938.2.2 Tlvresult . . . . .	928
8.939unpack_rms_GetSMSWake_t Struct Reference . . . . .	929
8.939.1 Detailed Description . . . . .	929
8.939.2 Field Documentation . . . . .	929
8.939.2.1 enabled . . . . .	929
8.939.2.2 ParamPresenceMask . . . . .	929

8.939.2.3 wake_mask . . . . .	929
8.940unpack_rms_SetSMSWake_t Struct Reference . . . . .	929
8.940.1 Detailed Description . . . . .	929
8.940.2 Field Documentation . . . . .	930
8.940.2.1 ParamPresenceMask . . . . .	930
8.941unpack_RMTransferStatistics_ind_t Struct Reference . . . . .	930
8.941.1 Detailed Description . . . . .	930
8.941.2 Field Documentation . . . . .	930
8.941.2.1 ParamPresenceMask . . . . .	930
8.941.2.2 RxDropConutTlv . . . . .	930
8.941.2.3 RxOkByteCountTlv . . . . .	931
8.941.2.4 RxOkConutTlv . . . . .	931
8.941.2.5 TxDropConutTlv . . . . .	931
8.941.2.6 TxOkByteCountTlv . . . . .	931
8.941.2.7 TxOkConutTlv . . . . .	931
8.942unpack_sar_SLQSGetRfSarState_t Struct Reference . . . . .	931
8.942.1 Detailed Description . . . . .	931
8.942.2 Field Documentation . . . . .	931
8.942.2.1 ParamPresenceMask . . . . .	931
8.942.2.2 pSarRFState . . . . .	931
8.943unpack_sms_GetSMSCAddress_t Struct Reference . . . . .	931
8.943.1 Detailed Description . . . . .	932
8.943.2 Field Documentation . . . . .	932
8.943.2.1 addressSize . . . . .	932
8.943.2.2 ParamPresenceMask . . . . .	932
8.943.2.3 pSMSCAddress . . . . .	932
8.943.2.4 pSMSCType . . . . .	932
8.943.2.5 typeSize . . . . .	932
8.944unpack_sms_SaveSMS_t Struct Reference . . . . .	932
8.944.1 Detailed Description . . . . .	933
8.944.2 Field Documentation . . . . .	933
8.944.2.1 ParamPresenceMask . . . . .	933
8.944.2.2 pMessageIndex . . . . .	933
8.945unpack_sms_SendSMS_t Struct Reference . . . . .	933
8.945.1 Detailed Description . . . . .	933
8.945.2 Field Documentation . . . . .	933
8.945.2.1 messageFailureCode . . . . .	933
8.945.2.2 messageId . . . . .	933
8.945.2.3 ParamPresenceMask . . . . .	933
8.946unpack_sms_SetNewSMSCallback_ind_t Struct Reference . . . . .	934

8.946.1 Detailed Description	934
8.946.2 Field Documentation	935
8.946.2.1 ETWSPLMNTlv	935
8.946.2.2 ETWSTlv	935
8.946.2.3 IMSTlv	935
8.946.2.4 MMTlv	935
8.946.2.5 NewMMTlv	935
8.946.2.6 ParamPresenceMask	935
8.946.2.7 SMSCTlv	935
8.946.2.8 TRMessageTlv	935
8.947unpack_sms_SetNewSMSCallback_t Struct Reference	935
8.947.1 Detailed Description	935
8.947.2 Field Documentation	935
8.947.2.1 ParamPresenceMask	935
8.948unpack_sms_SetSMSCAddress_t Struct Reference	935
8.948.1 Detailed Description	935
8.948.2 Field Documentation	936
8.948.2.1 ParamPresenceMask	936
8.949unpack_sms_SLQSDeleteSMS_t Struct Reference	936
8.949.1 Detailed Description	936
8.949.2 Field Documentation	936
8.949.2.1 ParamPresenceMask	936
8.950unpack_sms_SLQSGetIndicationRegister_t Struct Reference	936
8.950.1 Detailed Description	936
8.950.2 Field Documentation	937
8.950.2.1 ParamPresenceMask	937
8.950.2.2 pGetIndicationRegInfo	937
8.951unpack_sms_SLQSGetMessageWaiting_t Struct Reference	937
8.951.1 Detailed Description	937
8.951.2 Field Documentation	937
8.951.2.1 ParamPresenceMask	937
8.951.2.2 pGetMsgWaitingInfoResp	937
8.952unpack_sms_SLQSGetSMS_t Struct Reference	937
8.952.1 Detailed Description	937
8.952.2 Field Documentation	938
8.952.2.1 message	938
8.952.2.2 messageFormat	938
8.952.2.3 messageSize	938
8.952.2.4 messageTag	938
8.952.2.5 ParamPresenceMask	938

8.953unpack_sms_SLQSGetSmsBroadcastConfig_t Struct Reference . . . . .	938
8.953.1 Detailed Description . . . . .	939
8.953.2 Field Documentation . . . . .	939
8.953.2.1 ParamPresenceMask . . . . .	939
8.953.2.2 pBroadcastConfig . . . . .	939
8.953.2.3 pCDMABroadcastConfig . . . . .	939
8.954unpack_sms_SLQSGetSMSList_t Struct Reference . . . . .	939
8.954.1 Detailed Description . . . . .	939
8.954.2 Field Documentation . . . . .	940
8.954.2.1 messageList . . . . .	940
8.954.2.2 messageListSize . . . . .	940
8.954.2.3 ParamPresenceMask . . . . .	940
8.955unpack_sms_SLQSGetTransLayerInfo_t Struct Reference . . . . .	940
8.955.1 Detailed Description . . . . .	940
8.955.2 Field Documentation . . . . .	940
8.955.2.1 ParamPresenceMask . . . . .	940
8.955.2.2 pGetTransLayerInfo . . . . .	940
8.956unpack_sms_SLQSGetTransNWRegInfo_t Struct Reference . . . . .	940
8.956.1 Detailed Description . . . . .	941
8.956.2 Field Documentation . . . . .	941
8.956.2.1 ParamPresenceMask . . . . .	941
8.956.2.2 pGetTransNWRegInfo . . . . .	941
8.957unpack_sms_SLQSModifySMSStatus_t Struct Reference . . . . .	941
8.957.1 Detailed Description . . . . .	941
8.957.2 Field Documentation . . . . .	941
8.957.2.1 ParamPresenceMask . . . . .	941
8.958unpack_sms_SLQSNWRegInfoCallback_ind_t Struct Reference . . . . .	941
8.958.1 Detailed Description . . . . .	942
8.958.2 Field Documentation . . . . .	942
8.958.2.1 NWRegStat . . . . .	942
8.958.2.2 ParamPresenceMask . . . . .	942
8.959unpack_sms_SLQSSendAsyncSMS_t Struct Reference . . . . .	942
8.959.1 Detailed Description . . . . .	942
8.959.2 Field Documentation . . . . .	942
8.959.2.1 ParamPresenceMask . . . . .	942
8.960unpack_sms_SLQSSetIndicationRegister_t Struct Reference . . . . .	942
8.960.1 Detailed Description . . . . .	943
8.960.2 Field Documentation . . . . .	943
8.960.2.1 ParamPresenceMask . . . . .	943
8.961unpack_sms_SLQSSetSmsBroadcastActivation_t Struct Reference . . . . .	943



8.961.1 Detailed Description	943
8.961.2 Field Documentation	943
8.961.2.1 ParamPresenceMask	943
8.962unpack_sms_SLQSSetSmsBroadcastConfig_t Struct Reference	943
8.962.1 Detailed Description	943
8.962.2 Field Documentation	944
8.962.2.1 ParamPresenceMask	944
8.963unpack_sms_SLQSSetSmsStorage_t Struct Reference	944
8.963.1 Detailed Description	944
8.963.2 Field Documentation	944
8.963.2.1 ParamPresenceMask	944
8.964unpack_sms_SLQSSmsGetMaxStorageSize_t Struct Reference	944
8.964.1 Detailed Description	944
8.964.2 Field Documentation	944
8.964.2.1 ParamPresenceMask	945
8.964.2.2 pMaxStorageSizeResp	945
8.965unpack_sms_SLQSSmsGetMessageProtocol_t Struct Reference	945
8.965.1 Detailed Description	945
8.965.2 Field Documentation	945
8.965.2.1 ParamPresenceMask	945
8.965.2.2 pMessageProtocol	945
8.966unpack_sms_SLQSSmsSetRoutes_t Struct Reference	945
8.966.1 Detailed Description	945
8.966.2 Field Documentation	945
8.966.2.1 ParamPresenceMask	945
8.967unpack_sms_SLQSSwiGetSMSSStorage_t Struct Reference	946
8.967.1 Detailed Description	946
8.967.2 Field Documentation	946
8.967.2.1 ParamPresenceMask	946
8.967.2.2 pSmsStorage	946
8.968unpack_sms_SLQSTransLayerInfoCallback_ind_t Struct Reference	946
8.968.1 Detailed Description	946
8.968.2 Field Documentation	947
8.968.2.1 ParamPresenceMask	947
8.968.2.2 pTransLayerInfo	947
8.968.2.3 regInd	947
8.969unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t Struct Reference	947
8.969.1 Detailed Description	947
8.969.2 Field Documentation	949
8.969.2.1 alphaIDLen	949

8.969.2.2 causeCode . . . . .	949
8.969.2.3 errorClass . . . . .	949
8.969.2.4 messageId . . . . .	949
8.969.2.5 msgDelFailureCause . . . . .	949
8.969.2.6 msgDelFailureType . . . . .	949
8.969.2.7 pAlphaID . . . . .	949
8.969.2.8 ParamPresenceMask . . . . .	949
8.969.2.9 RPCause . . . . .	949
8.969.2.10 sendStatus . . . . .	949
8.969.2.11 TPCause . . . . .	949
8.969.2.12 userData . . . . .	949
8.970unpack_sms_SLQSWmsMemoryFullCallBack_ind_t Struct Reference . . . . .	949
8.970.1 Detailed Description . . . . .	949
8.970.2 Field Documentation . . . . .	950
8.970.2.1 messageMode . . . . .	950
8.970.2.2 ParamPresenceMask . . . . .	950
8.970.2.3 storageType . . . . .	950
8.971unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t Struct Reference . . . . .	950
8.971.1 Detailed Description . . . . .	950
8.971.2 Field Documentation . . . . .	950
8.971.2.1 msgWaitInfo . . . . .	950
8.971.2.2 numInstances . . . . .	950
8.971.2.3 ParamPresenceMask . . . . .	950
8.972unpack_swiaudio_SLQSGetM2MAudioProfile_t Struct Reference . . . . .	950
8.972.1 Detailed Description . . . . .	951
8.972.2 Field Documentation . . . . .	951
8.972.2.1 CwtMute . . . . .	951
8.972.2.2 EarMute . . . . .	951
8.972.2.3 Generator . . . . .	951
8.972.2.4 MicMute . . . . .	952
8.972.2.5 ParamPresenceMask . . . . .	952
8.972.2.6 Profile . . . . .	952
8.972.2.7 Volume . . . . .	952
8.973unpack_swiaudio_SLQSGetM2MAudioVolume_t Struct Reference . . . . .	952
8.973.1 Detailed Description . . . . .	952
8.973.2 Field Documentation . . . . .	952
8.973.2.1 Level . . . . .	952
8.973.2.2 ParamPresenceMask . . . . .	952
8.974unpack_swiaudio_SLQSGetM2MAVMute_t Struct Reference . . . . .	952
8.974.1 Detailed Description . . . . .	952

8.974.2 Field Documentation . . . . .	953
8.974.2.1 CwtMute . . . . .	953
8.974.2.2 EarMute . . . . .	953
8.974.2.3 MicMute . . . . .	953
8.974.2.4 ParamPresenceMask . . . . .	953
8.975unpack_swiaudio_SLQSGetM2MSpkrGain_t Struct Reference . . . . .	953
8.975.1 Detailed Description . . . . .	953
8.975.2 Field Documentation . . . . .	954
8.975.2.1 ParamPresenceMask . . . . .	954
8.975.2.2 Value . . . . .	954
8.976unpack_swiaavms_SLQSAVMSEventReportInd_t Struct Reference . . . . .	954
8.976.1 Detailed Description . . . . .	954
8.976.2 Field Documentation . . . . .	955
8.976.2.1 ParamPresenceMask . . . . .	955
8.976.2.2 pBinaryUpdateSessionInfo . . . . .	955
8.976.2.3 pConfig . . . . .	955
8.976.2.4 pConnectionRequest . . . . .	955
8.976.2.5 pDataSessionStatus . . . . .	955
8.976.2.6 pHTTPStatus . . . . .	955
8.976.2.7 pNotification . . . . .	955
8.976.2.8 pPackageID . . . . .	955
8.976.2.9 pRegStatus . . . . .	955
8.976.2.10pSessionType . . . . .	955
8.976.2.11pWAMSParaChanged . . . . .	955
8.976.2.12resultcode . . . . .	955
8.977unpack_swiaavms_SLQSAVMSSettings_t Struct Reference . . . . .	955
8.977.1 Detailed Description . . . . .	956
8.977.2 Field Documentation . . . . .	957
8.977.2.1 AutoConnect . . . . .	957
8.977.2.2 AutoReboot . . . . .	957
8.977.2.3 FwAutoSDM . . . . .	957
8.977.2.4 FwPromptdownload . . . . .	958
8.977.2.5 FwPromptUpdate . . . . .	958
8.977.2.6 OMADMEEnabled . . . . .	958
8.977.2.7 pAPNInfo . . . . .	958
8.977.2.8 ParamPresenceMask . . . . .	958
8.977.2.9 pConnectionRetryTimers . . . . .	958
8.977.2.10pNotificationStore . . . . .	958
8.977.2.11pPeroidsInfo . . . . .	958
8.977.2.12pPollingTimer . . . . .	958

8.977.2.13resultcode . . . . .	958
8.978unpack_swiaavms_SLQSAVMSGetSettings_v2_t Struct Reference . . . . .	958
8.978.1 Detailed Description . . . . .	958
8.978.2 Field Documentation . . . . .	960
8.978.2.1 AutoConnect . . . . .	960
8.978.2.2 FwAutoSDM . . . . .	960
8.978.2.3 FwPromptdownload . . . . .	960
8.978.2.4 FwPromptUpdate . . . . .	960
8.978.2.5 OMADMEEnabled . . . . .	960
8.978.2.6 pAPNInfo . . . . .	960
8.978.2.7 ParamPresenceMask . . . . .	960
8.978.2.8 pAutoReboot . . . . .	960
8.978.2.9 pConnectionRetryTimers . . . . .	960
8.978.2.10pNotificationStore . . . . .	960
8.978.2.11pPeroidsInfo . . . . .	960
8.978.2.12pPollingTimer . . . . .	960
8.978.2.13resultcode . . . . .	960
8.979unpack_swiaavms_SLQSAVMSSendSelection_t Struct Reference . . . . .	960
8.979.1 Detailed Description . . . . .	960
8.979.2 Field Documentation . . . . .	961
8.979.2.1 ParamPresenceMask . . . . .	961
8.979.2.2 resultcode . . . . .	961
8.980unpack_swiaavms_SLQSAVMSSessionGetInfo_t Struct Reference . . . . .	961
8.980.1 Detailed Description . . . . .	961
8.980.2 Field Documentation . . . . .	961
8.980.2.1 ParamPresenceMask . . . . .	961
8.980.2.2 pBinaryUpdateSessionInfo . . . . .	961
8.980.2.3 pConfig . . . . .	962
8.980.2.4 pNotification . . . . .	962
8.980.2.5 pPackageID . . . . .	962
8.980.2.6 resultcode . . . . .	962
8.981unpack_swiaavms_SLQSAvmsSetEventReport_t Struct Reference . . . . .	962
8.981.1 Detailed Description . . . . .	962
8.981.2 Field Documentation . . . . .	962
8.981.2.1 ParamPresenceMask . . . . .	962
8.981.2.2 resultcode . . . . .	962
8.982unpack_swiaavms_SLQSAVMSSetSettings_t Struct Reference . . . . .	962
8.982.1 Detailed Description . . . . .	962
8.982.2 Field Documentation . . . . .	962
8.982.2.1 ParamPresenceMask . . . . .	962

8.982.2.2 resultcode . . . . .	963
8.983unpack_swiaavms_SLQSAVMSSetSettings_v2_t Struct Reference . . . . .	963
8.983.1 Detailed Description . . . . .	963
8.983.2 Field Documentation . . . . .	963
8.983.2.1 ParamPresenceMask . . . . .	963
8.983.2.2 resultcode . . . . .	963
8.984unpack_swiaavms_SLQSAVMSStartSession_t Struct Reference . . . . .	963
8.984.1 Detailed Description . . . . .	963
8.984.2 Field Documentation . . . . .	963
8.984.2.1 ParamPresenceMask . . . . .	963
8.984.2.2 resultcode . . . . .	964
8.984.2.3 sessionResponse . . . . .	964
8.985unpack_swiaavms_SLQSAVMSStopSession_t Struct Reference . . . . .	964
8.985.1 Detailed Description . . . . .	964
8.985.2 Field Documentation . . . . .	964
8.985.2.1 ParamPresenceMask . . . . .	964
8.985.2.2 resultcode . . . . .	964
8.986unpack_swidms_SLQSSwiDmsGetHWWatchdog_t Struct Reference . . . . .	964
8.986.1 Detailed Description . . . . .	964
8.986.2 Field Documentation . . . . .	964
8.986.2.1 ParamPresenceMask . . . . .	964
8.986.2.2 pHWWatchdog . . . . .	965
8.987unpack_swidms_SLQSSwiDmsGetMTU_t Struct Reference . . . . .	965
8.987.1 Detailed Description . . . . .	965
8.987.2 Field Documentation . . . . .	965
8.987.2.1 ParamPresenceMask . . . . .	965
8.987.2.2 pEhrpdMTUSize . . . . .	965
8.987.2.3 pHrpdMTUSize . . . . .	965
8.987.2.4 pMTUSize3gpp . . . . .	965
8.987.2.5 pUsbMTUSize . . . . .	965
8.988unpack_swidms_SLQSSwiDmsGetSecureInfo_t Struct Reference . . . . .	965
8.988.1 Detailed Description . . . . .	966
8.988.2 Field Documentation . . . . .	966
8.988.2.1 jtagAccessAllowed . . . . .	966
8.988.2.2 memoryDumpAllowed . . . . .	966
8.988.2.3 ParamPresenceMask . . . . .	966
8.988.2.4 secureBootEnabled . . . . .	966
8.988.2.5 TlvResult . . . . .	966
8.989unpack_swidms_SLQSSwiDmsGetUsbComp_t Struct Reference . . . . .	966
8.989.1 Detailed Description . . . . .	967

8.989.2 Field Documentation . . . . .	967
8.989.2.1 ParamPresenceMask . . . . .	967
8.989.2.2 pInterfaceCfg . . . . .	967
8.989.2.3 pSupportedBitmasks . . . . .	967
8.990unpack_swidms_SLQSSwiDmsGetUsbNetNum_t Struct Reference . . . . .	967
8.990.1 Detailed Description . . . . .	967
8.990.2 Field Documentation . . . . .	967
8.990.2.1 ParamPresenceMask . . . . .	967
8.990.2.2 usbNetNum . . . . .	967
8.991unpack_swidms_SLQSSwiDmsSetHWWatchdog_t Struct Reference . . . . .	968
8.991.1 Detailed Description . . . . .	968
8.991.2 Field Documentation . . . . .	968
8.991.2.1 ParamPresenceMask . . . . .	968
8.991.2.2 Tlvresult . . . . .	968
8.992unpack_swidms_SLQSSwiDmsSetMTU_t Struct Reference . . . . .	968
8.992.1 Detailed Description . . . . .	968
8.992.2 Field Documentation . . . . .	968
8.992.2.1 ParamPresenceMask . . . . .	968
8.992.2.2 Tlvresult . . . . .	969
8.993unpack_swidms_SLQSSwiDmsSetUsbComp_t Struct Reference . . . . .	969
8.993.1 Detailed Description . . . . .	969
8.993.2 Field Documentation . . . . .	969
8.993.2.1 ParamPresenceMask . . . . .	969
8.993.2.2 Tlvresult . . . . .	969
8.994unpack_swiloc_SwiLocGetAutoStart_t Struct Reference . . . . .	969
8.994.1 Detailed Description . . . . .	969
8.994.2 Field Documentation . . . . .	971
8.994.2.1 fix_rate . . . . .	971
8.994.2.2 fix_rate_reported . . . . .	971
8.994.2.3 fix_type . . . . .	971
8.994.2.4 fix_type_reported . . . . .	971
8.994.2.5 function . . . . .	971
8.994.2.6 function_reported . . . . .	971
8.994.2.7 max_dist . . . . .	971
8.994.2.8 max_dist_reported . . . . .	971
8.994.2.9 max_time . . . . .	971
8.994.2.10max_time_reported . . . . .	971
8.994.2.11ParamPresenceMask . . . . .	971
8.995unpack_swioama_SLQSOMADMAAlertCallback_ind_t Struct Reference . . . . .	971
8.995.1 Detailed Description . . . . .	971

8.995.2 Field Documentation . . . . .	972
8.995.2.1 eventType . . . . .	972
8.995.2.2 ParamPresenceMask . . . . .	972
8.995.2.3 SessionInfoConfig . . . . .	972
8.995.2.4 SessionInfoFota . . . . .	972
8.995.2.5 SessionInfoNotification . . . . .	972
8.996unpack_swioma_SLQSOMADMGetSessionInfo_t Struct Reference . . . . .	972
8.996.1 Detailed Description . . . . .	972
8.996.2 Field Documentation . . . . .	974
8.996.2.1 Date . . . . .	974
8.996.2.2 DateLength . . . . .	975
8.996.2.3 ParamPresenceMask . . . . .	975
8.996.2.4 PkgDescLength . . . . .	975
8.996.2.5 PkgDescription . . . . .	975
8.996.2.6 PkgName . . . . .	975
8.996.2.7 PkgNameLength . . . . .	975
8.996.2.8 RetryCount . . . . .	975
8.996.2.9 SessionState . . . . .	975
8.996.2.10SessionType . . . . .	975
8.996.2.11Severity . . . . .	975
8.996.2.12Source . . . . .	975
8.996.2.13SourceLength . . . . .	975
8.996.2.14Status . . . . .	975
8.996.2.15Time . . . . .	975
8.996.2.16TimeLength . . . . .	975
8.996.2.17UpdateCompleteStatus . . . . .	975
8.997unpack_swioma_SLQSOMADMGetSessionInfoExt_t Struct Reference . . . . .	975
8.997.1 Detailed Description . . . . .	976
8.997.2 Field Documentation . . . . .	978
8.997.2.1 fumoResultCode . . . . .	978
8.997.2.2 fumoState . . . . .	978
8.997.2.3 hfaMaxRetry . . . . .	978
8.997.2.4 hfaRetryIndex . . . . .	978
8.997.2.5 hfaRetryInterval . . . . .	978
8.997.2.6 hfaStatus . . . . .	978
8.997.2.7 ParamPresenceMask . . . . .	978
8.997.2.8 pkgDate . . . . .	978
8.997.2.9 pkgDateLength . . . . .	978
8.997.2.10pkgDesc . . . . .	978
8.997.2.11pkgDescLength . . . . .	978

8.997.2.12pkgInstallTime . . . . .	978
8.997.2.13pkgInstallTimeLength . . . . .	978
8.997.2.14pkgName . . . . .	978
8.997.2.15pkgNameLength . . . . .	978
8.997.2.16pkgSize . . . . .	978
8.997.2.17pkgVendorName . . . . .	978
8.997.2.18pkgVendorNameLength . . . . .	978
8.997.2.19pkgVersionName . . . . .	978
8.997.2.20pkgVersionNameLength . . . . .	978
8.997.2.21sessionState . . . . .	978
8.997.2.22status . . . . .	978
8.998unpack_swioma_SLQSOMADMGetSettings_t Struct Reference . . . . .	979
8.998.1 Detailed Description . . . . .	979
8.998.2 Field Documentation . . . . .	980
8.998.2.1 Autosdm . . . . .	980
8.998.2.2 FOTAdownload . . . . .	980
8.998.2.3 FOTAUpdate . . . . .	980
8.998.2.4 FwAutoCheck . . . . .	980
8.998.2.5 OMADMEEnabled . . . . .	980
8.998.2.6 ParamPresenceMask . . . . .	980
8.999unpack_swioma_SLQSOMADMStartSession_t Struct Reference . . . . .	980
8.999.1 Detailed Description . . . . .	980
8.999.2 Field Documentation . . . . .	981
8.999.2.1 FwAvailability . . . . .	981
8.999.2.2 ParamPresenceMask . . . . .	981
8.1000unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t Struct Reference . . . . .	981
8.1000.1 Detailed Description . . . . .	981
8.1000.2 Field Documentation . . . . .	981
8.1000.2.1ParamPresenceMask . . . . .	981
8.1000.2.2Tlvresult . . . . .	981
8.1001unpack_tmd_SLQSTmdGetMitigationDevList_t Struct Reference . . . . .	981
8.1001.1 Detailed Description . . . . .	981
8.1001.2 Field Documentation . . . . .	982
8.1001.2.1MitigationDevList . . . . .	982
8.1001.2.2MitigationDevListLen . . . . .	982
8.1001.2.3ParamPresenceMask . . . . .	982
8.1001.2.4Tlvresult . . . . .	982
8.1002unpack_tmd_SLQSTmdGetMitigationLvl_t Struct Reference . . . . .	982
8.1002.1 Detailed Description . . . . .	982
8.1002.2 Field Documentation . . . . .	983



8.1002.2.1CurrentmitigationLvl . . . . .	983
8.1002.2.2ParamPresenceMask . . . . .	983
8.1002.2.3ReqMitigationLvl . . . . .	983
8.1002.2.4Tlvresult . . . . .	983
8.1003 <del>8</del> unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t Struct Reference . . . . .	983
8.1003.1Detailed Description . . . . .	983
8.1003.2Field Documentation . . . . .	983
8.1003.2.1deviceId . . . . .	983
8.1003.2.2deviceIdLen . . . . .	983
8.1003.2.3vl . . . . .	983
8.1003.2.4ParamPresenceMask . . . . .	983
8.1004 <del>4</del> unpack_tmd_SLQSTmdRegNotMitigationLvl_t Struct Reference . . . . .	983
8.1004.1Detailed Description . . . . .	984
8.1004.2Field Documentation . . . . .	984
8.1004.2.1ParamPresenceMask . . . . .	984
8.1004.2.2Tlvresult . . . . .	984
8.1005 <del>5</del> unpack_uim_ChangePin_t Struct Reference . . . . .	984
8.1005.1Detailed Description . . . . .	984
8.1005.2Field Documentation . . . . .	985
8.1005.2.1ParamPresenceMask . . . . .	985
8.1005.2.2pEncryptedPIN1 . . . . .	985
8.1005.2.3pIndicationToken . . . . .	985
8.1005.2.4pRemainingRetries . . . . .	985
8.1005.2.5Tlvresult . . . . .	985
8.1006 <del>6</del> unpack_uim_GetCardStatus_t Struct Reference . . . . .	985
8.1006.1Detailed Description . . . . .	985
8.1006.2Field Documentation . . . . .	985
8.1006.2.1ParamPresenceMask . . . . .	985
8.1006.2.2pCardStatus . . . . .	985
8.1006.2.3pHotSwapStatus . . . . .	985
8.1006.2.4Tlvresult . . . . .	986
8.1007 <del>7</del> unpack_uim_GetCardStatusV2_t Struct Reference . . . . .	986
8.1007.1Detailed Description . . . . .	986
8.1007.2Field Documentation . . . . .	986
8.1007.2.1ParamPresenceMask . . . . .	986
8.1007.2.2pCardStatus . . . . .	986
8.1007.2.3pHotSwapStatus . . . . .	986
8.1007.2.4pSimBusyStatus . . . . .	986
8.1007.2.5pValidCardStatus . . . . .	986
8.1007.2.6Tlvresult . . . . .	986

8.1008	<a href="#">npack_uim_ReadTransparent_t Struct Reference</a>	987
8.1008.1	<a href="#">Detailed Description</a>	987
8.1008.2	<a href="#">Field Documentation</a>	987
8.1008.2.1	<a href="#">ParamPresenceMask</a>	987
8.1008.2.2	<a href="#">pCardResult</a>	987
8.1008.2.3	<a href="#">pEncryptedData</a>	987
8.1008.2.4	<a href="#">pIndicationToken</a>	987
8.1008.2.5	<a href="#">pReadResult</a>	987
8.1008.2.6	<a href="#">Tlvresult</a>	987
8.1009	<a href="#">npack_uim_SetPinProtection_t Struct Reference</a>	988
8.1009.1	<a href="#">Detailed Description</a>	988
8.1009.2	<a href="#">Field Documentation</a>	988
8.1009.2.1	<a href="#">ParamPresenceMask</a>	988
8.1009.2.2	<a href="#">pEncryptedPIN1</a>	988
8.1009.2.3	<a href="#">pIndicationToken</a>	988
8.1009.2.4	<a href="#">pRemainingRetries</a>	988
8.1009.2.5	<a href="#">Tlvresult</a>	988
8.1010	<a href="#">npack_uim_SetUimSlotStatusChangeCallback_ind_t Struct Reference</a>	988
8.1010.1	<a href="#">Detailed Description</a>	989
8.1010.2	<a href="#">Field Documentation</a>	989
8.1010.2.1	<a href="#">bNumberOfPhySlots</a>	989
8.1010.2.2	<a href="#">ParamPresenceMask</a>	989
8.1010.2.3	<a href="#">slotsstatusChange</a>	989
8.1011	<a href="#">npack_uim_SLQSUIMAuthenticate_t Struct Reference</a>	989
8.1011.1	<a href="#">Detailed Description</a>	989
8.1011.2	<a href="#">Field Documentation</a>	990
8.1011.2.1	<a href="#">ParamPresenceMask</a>	990
8.1011.2.2	<a href="#">pAuthenticateResult</a>	990
8.1011.2.3	<a href="#">pCardResult</a>	990
8.1011.2.4	<a href="#">pIndicationToken</a>	990
8.1012	<a href="#">npack_uim_SLQSUIMDepersonalization_t Struct Reference</a>	990
8.1012.1	<a href="#">Detailed Description</a>	990
8.1012.2	<a href="#">Field Documentation</a>	990
8.1012.2.1	<a href="#">ParamPresenceMask</a>	990
8.1012.2.2	<a href="#">pRemainingRetries</a>	990
8.1013	<a href="#">npack_uim_SLQSUIEventRegister_t Struct Reference</a>	990
8.1013.1	<a href="#">Detailed Description</a>	990
8.1013.2	<a href="#">Field Documentation</a>	991
8.1013.2.1	<a href="#">eventMask</a>	991
8.1013.2.2	<a href="#">ParamPresenceMask</a>	991

8.1014	<a href="#">unpack_uim_SLQSUIMGetConfiguration_t Struct Reference</a>	991
8.1014.1	<a href="#">Detailed Description</a>	991
8.1014.2	<a href="#">Field Documentation</a>	992
8.1014.2.1	<a href="#">ParamPresenceMask</a>	992
8.1014.2.2	<a href="#">pAutoSelection</a>	992
8.1014.2.3	<a href="#">pHaltSubscription</a>	992
8.1014.2.4	<a href="#">pPersonalizationStatus</a>	992
8.1015	<a href="#">unpack_uim_SLQSUIMGetFileAttributes_t Struct Reference</a>	992
8.1015.1	<a href="#">Detailed Description</a>	992
8.1015.2	<a href="#">Field Documentation</a>	993
8.1015.2.1	<a href="#">ParamPresenceMask</a>	993
8.1015.2.2	<a href="#">pCardResult</a>	993
8.1015.2.3	<a href="#">pFileAttributes</a>	993
8.1015.2.4	<a href="#">pIndicationToken</a>	993
8.1016	<a href="#">unpack_uim_SLQSUIMGetServiceStatus_t Struct Reference</a>	993
8.1016.1	<a href="#">Detailed Description</a>	993
8.1016.2	<a href="#">Field Documentation</a>	993
8.1016.2.1	<a href="#">ParamPresenceMask</a>	993
8.1016.2.2	<a href="#">pFDNStatus</a>	993
8.1016.2.3	<a href="#">pHiddenKeyStatus</a>	993
8.1016.2.4	<a href="#">pIndex</a>	993
8.1017	<a href="#">unpack_uim_SLQSUIMGetSlotsStatus_t Struct Reference</a>	994
8.1017.1	<a href="#">Detailed Description</a>	994
8.1017.2	<a href="#">Field Documentation</a>	994
8.1017.2.1	<a href="#">ParamPresenceMask</a>	994
8.1017.2.2	<a href="#">pNumberOfPhySlot</a>	994
8.1017.2.3	<a href="#">pUimSlotsStatus</a>	994
8.1018	<a href="#">unpack_uim_SLQSUIMGetSlotsStatusV2_t Struct Reference</a>	994
8.1018.1	<a href="#">Detailed Description</a>	994
8.1018.2	<a href="#">Field Documentation</a>	995
8.1018.2.1	<a href="#">ParamPresenceMask</a>	995
8.1018.2.2	<a href="#">pGetSlotsInfoTlv</a>	995
8.1018.2.3	<a href="#">pGetSlotsStatusTlv</a>	995
8.1019	<a href="#">unpack_uim_SLQSUIMReadRecord_t Struct Reference</a>	995
8.1019.1	<a href="#">Detailed Description</a>	995
8.1019.2	<a href="#">Field Documentation</a>	996
8.1019.2.1	<a href="#">pAdditionalReadResult</a>	996
8.1019.2.2	<a href="#">ParamPresenceMask</a>	996
8.1019.2.3	<a href="#">pCardResult</a>	996
8.1019.2.4	<a href="#">pIndicationToken</a>	996

8.1019.2.5pReadResult . . . . .	996
8.1020unpack_uim_SLQSUIMRefreshCallback_Ind_t Struct Reference . . . . .	996
8.1020.1Detailed Description . . . . .	996
8.1020.2Field Documentation . . . . .	996
8.1020.2.1ParamPresenceMask . . . . .	996
8.1020.2.2RefreshEvent . . . . .	996
8.1020.2.3TlvPresent . . . . .	997
8.1021unpack_uim_SLQSUIMRefreshGetLastEvent_t Struct Reference . . . . .	997
8.1021.1Detailed Description . . . . .	997
8.1021.2Field Documentation . . . . .	997
8.1021.2.1ParamPresenceMask . . . . .	997
8.1021.2.2RefreshEvent . . . . .	997
8.1022unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t Struct Reference . . . . .	997
8.1022.1Detailed Description . . . . .	997
8.1022.2Field Documentation . . . . .	997
8.1022.2.1ParamPresenceMask . . . . .	998
8.1022.2.2CardStatus . . . . .	998
8.1023unpack_uim_SLQSUIMWriteRecord_t Struct Reference . . . . .	998
8.1023.1Detailed Description . . . . .	998
8.1023.2Field Documentation . . . . .	998
8.1023.2.1ParamPresenceMask . . . . .	998
8.1023.2.2CardResult . . . . .	998
8.1023.2.3IndicationToken . . . . .	998
8.1024unpack_uim_SLQSUIMWriteTransparent_t Struct Reference . . . . .	998
8.1024.1Detailed Description . . . . .	998
8.1024.2Field Documentation . . . . .	999
8.1024.2.1ParamPresenceMask . . . . .	999
8.1024.2.2CardResult . . . . .	999
8.1024.2.3IndicationToken . . . . .	999
8.1025unpack_uim_UnblockPin_t Struct Reference . . . . .	999
8.1025.1Detailed Description . . . . .	999
8.1025.2Field Documentation . . . . .	1000
8.1025.2.1ParamPresenceMask . . . . .	1000
8.1025.2.2EncryptedPIN1 . . . . .	1000
8.1025.2.3IndicationToken . . . . .	1000
8.1025.2.4RemainingRetries . . . . .	1000
8.1025.2.5Tlvresult . . . . .	1000
8.1026unpack_uim_UnblockPinV2_t Struct Reference . . . . .	1000
8.1026.1Detailed Description . . . . .	1000
8.1026.2Field Documentation . . . . .	1001

8.1026.2.1ParamPresenceMask . . . . .	1001
8.1026.2.2pCardResult . . . . .	1001
8.1026.2.3pEncryptedPIN1 . . . . .	1001
8.1026.2.4pIndicationToken . . . . .	1001
8.1026.2.5pRemainingRetries . . . . .	1001
8.1026.2.6Tlvresult . . . . .	1001
8.1027unpack_uim_VerifyPin_t Struct Reference . . . . .	1001
8.1027.1Detailed Description . . . . .	1001
8.1027.2Field Documentation . . . . .	1001
8.1027.2.1ParamPresenceMask . . . . .	1001
8.1027.2.2pEncryptedPIN1 . . . . .	1001
8.1027.2.3pIndicationToken . . . . .	1002
8.1027.2.4pRemainingRetries . . . . .	1002
8.1027.2.5Tlvresult . . . . .	1002
8.1028unpack_voice_allCallStatusCallback_ind_t Struct Reference . . . . .	1002
8.1028.1Detailed Description . . . . .	1002
8.1028.2Field Documentation . . . . .	1003
8.1028.2.1arrCallInfomation . . . . .	1003
8.1028.2.2ParamPresenceMask . . . . .	1003
8.1028.2.3pArrAlertingPattern . . . . .	1003
8.1028.2.4pArrAlertingType . . . . .	1003
8.1028.2.5pArrAlphaID . . . . .	1003
8.1028.2.6pArrCalledPartyNum . . . . .	1003
8.1028.2.7pArrCallEndReason . . . . .	1003
8.1028.2.8pArrConnectPartyNum . . . . .	1003
8.1028.2.9pArrDiagInfo . . . . .	1003
8.1028.2.10pArrRedirPartyNum . . . . .	1003
8.1028.2.11pArrRemotePartyName . . . . .	1003
8.1028.2.12pArrRemotePartyNum . . . . .	1003
8.1028.2.13pArrSvcOption . . . . .	1004
8.1029unpack_voice_DTMFEventCallback_ind_t Struct Reference . . . . .	1004
8.1029.1Detailed Description . . . . .	1004
8.1029.2Field Documentation . . . . .	1004
8.1029.2.1DTMFInformation . . . . .	1004
8.1029.2.2ParamPresenceMask . . . . .	1005
8.1029.2.3pOffLength . . . . .	1005
8.1029.2.4pOnLength . . . . .	1005
8.1030unpack_voice_OTASPStatusCallback_ind_t Struct Reference . . . . .	1005
8.1030.1Detailed Description . . . . .	1005
8.1030.2Field Documentation . . . . .	1006

8.1030.2.1callID . . . . .	1006
8.1030.2.2OTASPStatus . . . . .	1006
8.1030.2.3ParamPresenceMask . . . . .	1006
8.1031unpack_voice_SLQSOriginateUSSD_t Struct Reference . . . . .	1006
8.1031.1Detailed Description . . . . .	1006
8.1031.2Field Documentation . . . . .	1009
8.1031.2.1pAlphaIDInfo . . . . .	1009
8.1031.2.2ParamPresenceMask . . . . .	1009
8.1031.2.3pCallID . . . . .	1009
8.1031.2.4pCcResultType . . . . .	1009
8.1031.2.5pCCSuppsType . . . . .	1009
8.1031.2.6pfailureCause . . . . .	1009
8.1031.2.7pUSSDInfo . . . . .	1009
8.1032unpack_voice_SLQSVoiceAnswerCall_t Struct Reference . . . . .	1009
8.1032.1Detailed Description . . . . .	1009
8.1032.2Field Documentation . . . . .	1010
8.1032.2.1ParamPresenceMask . . . . .	1010
8.1032.2.2pCallID . . . . .	1010
8.1033unpack_voice_SLQSVoiceBurstDTMF_t Struct Reference . . . . .	1010
8.1033.1Detailed Description . . . . .	1010
8.1033.2Field Documentation . . . . .	1010
8.1033.2.1ParamPresenceMask . . . . .	1010
8.1033.2.2pCallID . . . . .	1010
8.1034unpack_voice_SLQSVoiceDialCall_t Struct Reference . . . . .	1010
8.1034.1Detailed Description . . . . .	1011
8.1034.2Field Documentation . . . . .	1011
8.1034.2.1pAlphaIDInfo . . . . .	1011
8.1034.2.2ParamPresenceMask . . . . .	1011
8.1034.2.3pCallID . . . . .	1011
8.1034.2.4pCCResultType . . . . .	1011
8.1034.2.5pCCSUPSType . . . . .	1011
8.1035unpack_voice_SLQSVoiceEndCall_t Struct Reference . . . . .	1011
8.1035.1Detailed Description . . . . .	1012
8.1035.2Field Documentation . . . . .	1012
8.1035.2.1ParamPresenceMask . . . . .	1012
8.1035.2.2pCallID . . . . .	1012
8.1036unpack_voice_SLQSVoiceGetAllCallInfo_t Struct Reference . . . . .	1012
8.1036.1Detailed Description . . . . .	1012
8.1036.2Field Documentation . . . . .	1015
8.1036.2.1ParamPresenceMask . . . . .	1015

8.1036.2.2pArrAlertingPattern . . . . .	1015
8.1036.2.3pArrAlertingType . . . . .	1015
8.1036.2.4pArrAlphaID . . . . .	1015
8.1036.2.5pArrCalledPartyNum . . . . .	1015
8.1036.2.6pArrCallEndReason . . . . .	1015
8.1036.2.7pArrCallInfo . . . . .	1015
8.1036.2.8pArrConnectPartyNum . . . . .	1015
8.1036.2.9pArrDiagInfo . . . . .	1015
8.1036.2.10pArrRedirPartyNum . . . . .	1015
8.1036.2.11pArrRemotePartyName . . . . .	1015
8.1036.2.12pArrRemotePartyNum . . . . .	1015
8.1036.2.13pArrSvcOption . . . . .	1015
8.1036.2.14pArrUUSInfo . . . . .	1015
8.1036.2.15OTASPStatus . . . . .	1015
8.1036.2.16VoicePrivacy . . . . .	1015
8.1037npack_voice_SLQSVoiceGetCallBarring_t Struct Reference . . . . .	1016
8.1037.1Detailed Description . . . . .	1016
8.1037.2Field Documentation . . . . .	1017
8.1037.2.1pAlphaIDInfo . . . . .	1017
8.1037.2.2ParamPresenceMask . . . . .	1017
8.1037.2.3pCallID . . . . .	1017
8.1037.2.4pCCResType . . . . .	1017
8.1037.2.5pCCSUPSType . . . . .	1017
8.1037.2.6pFailCause . . . . .	1017
8.1037.2.7pSvcClass . . . . .	1017
8.1038npack_voice_SLQSVoiceGetCallForwardingStatus_t Struct Reference . . . . .	1017
8.1038.1Detailed Description . . . . .	1017
8.1038.2Field Documentation . . . . .	1018
8.1038.2.1pAlphaIDInfo . . . . .	1018
8.1038.2.2ParamPresenceMask . . . . .	1018
8.1038.2.3pCallID . . . . .	1018
8.1038.2.4pCCResType . . . . .	1018
8.1038.2.5pCCSUPSType . . . . .	1018
8.1038.2.6pFailCause . . . . .	1019
8.1038.2.7pGetCallFWExtInfo . . . . .	1019
8.1038.2.8pGetCallFWInfo . . . . .	1019
8.1039npack_voice_SLQSVoiceGetCallInfo_t Struct Reference . . . . .	1019
8.1039.1Detailed Description . . . . .	1019
8.1039.2Field Documentation . . . . .	1021
8.1039.2.1pAlertingPattern . . . . .	1021

8.1039.2.2pAlertType . . . . .	1021
8.1039.2.3pAlphaIDInfo . . . . .	1021
8.1039.2.4ParamPresenceMask . . . . .	1021
8.1039.2.5pCallInfo . . . . .	1021
8.1039.2.6pConnectNumInfo . . . . .	1021
8.1039.2.7pDiagInfo . . . . .	1021
8.1039.2.8pOTASPSStatus . . . . .	1021
8.1039.2.9pRemotePartyName . . . . .	1021
8.1039.2.10pRemotePartyNum . . . . .	1021
8.1039.2.11pSrvOpt . . . . .	1022
8.1039.2.12pJUSInfo . . . . .	1022
8.1039.2.13pVoicePrivacy . . . . .	1022
8.1040unpack_voice_SLQSVoiceGetCallWaiting_t Struct Reference . . . . .	1022
8.1040.1Detailed Description . . . . .	1022
8.1040.2Field Documentation . . . . .	1023
8.1040.2.1pAlphaIDInfo . . . . .	1023
8.1040.2.2ParamPresenceMask . . . . .	1023
8.1040.2.3pCallID . . . . .	1023
8.1040.2.4pCCResType . . . . .	1023
8.1040.2.5pCCSUPSType . . . . .	1023
8.1040.2.6pFailCause . . . . .	1023
8.1040.2.7pSvcClass . . . . .	1023
8.1041unpack_voice_SLQSVoiceGetCLIP_t Struct Reference . . . . .	1023
8.1041.1Detailed Description . . . . .	1023
8.1041.2Field Documentation . . . . .	1024
8.1041.2.1pAlphaIDInfo . . . . .	1024
8.1041.2.2ParamPresenceMask . . . . .	1024
8.1041.2.3pCallID . . . . .	1024
8.1041.2.4pCCResType . . . . .	1024
8.1041.2.5pCCSUPSType . . . . .	1024
8.1041.2.6pCLIPResp . . . . .	1025
8.1041.2.7pFailCause . . . . .	1025
8.1042unpack_voice_SLQSVoiceGetCLIR_t Struct Reference . . . . .	1025
8.1042.1Detailed Description . . . . .	1025
8.1042.2Field Documentation . . . . .	1026
8.1042.2.1pAlphaIDInfo . . . . .	1026
8.1042.2.2ParamPresenceMask . . . . .	1026
8.1042.2.3pCallID . . . . .	1026
8.1042.2.4pCCResType . . . . .	1026
8.1042.2.5pCCSUPSType . . . . .	1026



8.1042.2.6pCLIRResp . . . . .	1026
8.1042.2.7pFailCause . . . . .	1026
8.1043npack_voice_SLQSVoiceGetCNAP_t Struct Reference . . . . .	1026
8.1043.1Detailed Description . . . . .	1026
8.1043.2Field Documentation . . . . .	1027
8.1043.2.1pAlphaIDInfo . . . . .	1027
8.1043.2.2ParamPresenceMask . . . . .	1027
8.1043.2.3pCallID . . . . .	1027
8.1043.2.4pCCRResType . . . . .	1027
8.1043.2.5pCCSUPSType . . . . .	1027
8.1043.2.6pCNAPResp . . . . .	1027
8.1043.2.7pFailCause . . . . .	1027
8.1044npack_voice_SLQSVoiceGetCOLP_t Struct Reference . . . . .	1028
8.1044.1Detailed Description . . . . .	1028
8.1044.2Field Documentation . . . . .	1029
8.1044.2.1pAlphaIDInfo . . . . .	1029
8.1044.2.2ParamPresenceMask . . . . .	1029
8.1044.2.3pCallID . . . . .	1029
8.1044.2.4pCCRResType . . . . .	1029
8.1044.2.5pCCSUPSType . . . . .	1029
8.1044.2.6pCOLPResp . . . . .	1029
8.1044.2.7pFailCause . . . . .	1029
8.1045npack_voice_SLQSVoiceGetCOLR_t Struct Reference . . . . .	1029
8.1045.1Detailed Description . . . . .	1029
8.1045.2Field Documentation . . . . .	1030
8.1045.2.1pAlphaIDInfo . . . . .	1030
8.1045.2.2ParamPresenceMask . . . . .	1030
8.1045.2.3pCallID . . . . .	1030
8.1045.2.4pCCRResType . . . . .	1030
8.1045.2.5pCCSUPSType . . . . .	1030
8.1045.2.6pCOLRResp . . . . .	1030
8.1045.2.7pFailCause . . . . .	1030
8.1046npack_voice_SLQSVoiceGetConfig_t Struct Reference . . . . .	1030
8.1046.1Detailed Description . . . . .	1031
8.1046.2Field Documentation . . . . .	1032
8.1046.2.1pAirTimerCnt . . . . .	1032
8.1046.2.2ParamPresenceMask . . . . .	1032
8.1046.2.3pAutoAnswerStat . . . . .	1032
8.1046.2.4pCurAMRConfig . . . . .	1032
8.1046.2.5pCurPrefVoiceSO . . . . .	1032

8.1046.2.6pCurrTTYMode . . . . .	1032
8.1046.2.7pCurVoiceDomainPref . . . . .	1032
8.1046.2.8pCurVoicePrivacyPref . . . . .	1032
8.1046.2.9pRoamTimerCnt . . . . .	1032
8.1047unpack_voice_SLQSVoiceManageCalls_t Struct Reference . . . . .	1032
8.1047.1Detailed Description . . . . .	1033
8.1047.2Field Documentation . . . . .	1033
8.1047.2.1ParamPresenceMask . . . . .	1033
8.1047.2.2pFailCause . . . . .	1033
8.1048unpack_voice_SLQSVoiceSendFlash_t Struct Reference . . . . .	1033
8.1048.1Detailed Description . . . . .	1033
8.1048.2Field Documentation . . . . .	1033
8.1048.2.1ParamPresenceMask . . . . .	1033
8.1048.2.2pCallID . . . . .	1034
8.1049unpack_voice_SLQSVoiceSetCallBarringPassword_t Struct Reference . . . . .	1034
8.1049.1Detailed Description . . . . .	1034
8.1049.2Field Documentation . . . . .	1035
8.1049.2.1pAlphaIDInfo . . . . .	1035
8.1049.2.2ParamPresenceMask . . . . .	1035
8.1049.2.3pCallID . . . . .	1035
8.1049.2.4pCCResType . . . . .	1035
8.1049.2.5pCCSUPSType . . . . .	1035
8.1049.2.6pFailCause . . . . .	1035
8.1050unpack_voice_SLQSVoiceSetConfig_t Struct Reference . . . . .	1035
8.1050.1Detailed Description . . . . .	1035
8.1050.2Field Documentation . . . . .	1036
8.1050.2.1pAirTimerStatus . . . . .	1036
8.1050.2.2ParamPresenceMask . . . . .	1036
8.1050.2.3pAutoAnsStatus . . . . .	1036
8.1050.2.4pPrefVoiceSOSStatus . . . . .	1036
8.1050.2.5pRoamTimerStatus . . . . .	1036
8.1050.2.6pTTYConfigStatus . . . . .	1036
8.1050.2.7pVoiceDomainPrefStatus . . . . .	1036
8.1051unpack_voice_SLQSVoiceSetSUPSService_t Struct Reference . . . . .	1037
8.1051.1Detailed Description . . . . .	1037
8.1051.2Field Documentation . . . . .	1038
8.1051.2.1pAlphaIDInfo . . . . .	1038
8.1051.2.2ParamPresenceMask . . . . .	1038
8.1051.2.3pCallID . . . . .	1038
8.1051.2.4pCCResultType . . . . .	1038

8.1051.2.5pCCSUPSType . . . . .	1038
8.1051.2.6pFailCause . . . . .	1038
8.1052npack_voice_SLQSVoiceStartContDTMF_t Struct Reference . . . . .	1038
8.1052.1Detailed Description . . . . .	1038
8.1052.2Field Documentation . . . . .	1038
8.1052.2.1ParamPresenceMask . . . . .	1038
8.1052.2.2pCallID . . . . .	1038
8.1053npack_voice_SLQSVoiceStopContDTMF_t Struct Reference . . . . .	1038
8.1053.1Detailed Description . . . . .	1039
8.1053.2Field Documentation . . . . .	1039
8.1053.2.1callID . . . . .	1039
8.1053.2.2ParamPresenceMask . . . . .	1039
8.1054npack_voice_SLQSVoiceSUPSCallback_ind_t Struct Reference . . . . .	1039
8.1054.1Detailed Description . . . . .	1039
8.1054.2Field Documentation . . . . .	1041
8.1054.2.1pAlphaIDInfo . . . . .	1041
8.1054.2.2ParamPresenceMask . . . . .	1041
8.1054.2.3pCallBarPasswd . . . . .	1041
8.1054.2.4pCallFwdInfo . . . . .	1042
8.1054.2.5pCallFWNum . . . . .	1042
8.1054.2.6pCallFWTimerVal . . . . .	1042
8.1054.2.7pCallID . . . . .	1042
8.1054.2.8pCLIPstatus . . . . .	1042
8.1054.2.9pCLIRstatus . . . . .	1042
8.1054.2.10pCNAPstatus . . . . .	1042
8.1054.2.11pCOLPstatus . . . . .	1042
8.1054.2.12pCOLRstatus . . . . .	1042
8.1054.2.13pDataSrc . . . . .	1042
8.1054.2.14pFailCause . . . . .	1042
8.1054.2.15pNewPwdData . . . . .	1042
8.1054.2.16pReason . . . . .	1042
8.1054.2.17pSvcClass . . . . .	1042
8.1054.2.18pUSSInfo . . . . .	1042
8.1054.2.19SUPSInformation . . . . .	1042
8.1055npack_voice_SUPSNotificationCallback_ind_t Struct Reference . . . . .	1042
8.1055.1Detailed Description . . . . .	1042
8.1055.2Field Documentation . . . . .	1044
8.1055.2.1callID . . . . .	1044
8.1055.2.2notifType . . . . .	1044
8.1055.2.3ParamPresenceMask . . . . .	1044

8.1055.2.4	pCUGIndex	1044
8.1055.2.5	pECTNum	1044
8.1056	unpack_voice_USSDNotificationCallback_ind_t Struct Reference	1044
8.1056.1	Detailed Description	1044
8.1056.2	Field Documentation	1044
8.1056.2.1	notification_Type	1044
8.1056.2.2	ParamPresenceMask	1044
8.1056.2.3	USSDNotificationNetworkInfo	1044
8.1057	unpack_voice_VoiceInfoRecCallback_ind_t Struct Reference	1044
8.1057.1	Detailed Description	1045
8.1057.2	Field Documentation	1047
8.1057.2.1	callID	1047
8.1057.2.2	ParamPresenceMask	1047
8.1057.2.3	CalledPartyInfo	1047
8.1057.2.4	CallerIDInfo	1047
8.1057.2.5	CallerNameInfo	1047
8.1057.2.6	CallingPartyInfo	1047
8.1057.2.7	CallWaitInd	1047
8.1057.2.8	CLIRCause	1047
8.1057.2.9	ConnectNumInfo	1047
8.1057.2.10	DispInfo	1047
8.1057.2.11	ExtDispInfo	1047
8.1057.2.12	ExtDispRecInfo	1047
8.1057.2.13	LineCtrlInfo	1047
8.1057.2.14	NSSAudioCtrl	1047
8.1057.2.15	NSSRelease	1047
8.1057.2.16	RedirNumInfo	1047
8.1057.2.17	SignalInfo	1047
8.1058	unpack_voice_voicePrivacyChangeCallback_ind_t Struct Reference	1047
8.1058.1	Detailed Description	1047
8.1058.2	Field Documentation	1048
8.1058.2.1	callID	1048
8.1058.2.2	ParamPresenceMask	1048
8.1058.2.3	VoicePrivacy	1048
8.1059	unpack_wds_DHCPv4ClientLease_ind_t Struct Reference	1048
8.1059.1	Detailed Description	1048
8.1059.2	Field Documentation	1049
8.1059.2.1	DHCPv4LeaseOptTlv	1049
8.1059.2.2	DHCPv4LeaseStateTlv	1049
8.1059.2.3	Pv4AddrTlv	1049

8.1059.2.4ParamPresenceMask . . . . .	1049
8.1059.2.5ProfileIdTlv . . . . .	1049
8.1060unpack_wds_GetAutoconnect_t Struct Reference . . . . .	1049
8.1060.1Detailed Description . . . . .	1049
8.1060.2Field Documentation . . . . .	1049
8.1060.2.1ParamPresenceMask . . . . .	1049
8.1060.2.2psetting . . . . .	1049
8.1061unpack_wds_GetByteTotals_t Struct Reference . . . . .	1049
8.1061.1Detailed Description . . . . .	1049
8.1061.2Field Documentation . . . . .	1050
8.1061.2.1ParamPresenceMask . . . . .	1050
8.1061.2.2pRXTotalBytes . . . . .	1050
8.1061.2.3pTXTotalBytes . . . . .	1050
8.1062unpack_wds_GetConnectionRate_t Struct Reference . . . . .	1050
8.1062.1Detailed Description . . . . .	1050
8.1062.2Field Documentation . . . . .	1051
8.1062.2.1currentChannelRXRate . . . . .	1051
8.1062.2.2currentChannelTXRate . . . . .	1051
8.1062.2.3maxChannelRXRate . . . . .	1051
8.1062.2.4maxChannelTXRate . . . . .	1051
8.1062.2.5ParamPresenceMask . . . . .	1051
8.1063unpack_wds_GetDataBearerTechnology_t Struct Reference . . . . .	1051
8.1063.1Detailed Description . . . . .	1051
8.1063.2Field Documentation . . . . .	1052
8.1063.2.1ParamPresenceMask . . . . .	1052
8.1063.2.2pDataBearer . . . . .	1052
8.1064unpack_wds_GetDefaultProfile_t Struct Reference . . . . .	1052
8.1064.1Detailed Description . . . . .	1053
8.1064.2Field Documentation . . . . .	1054
8.1064.2.1apnname . . . . .	1054
8.1064.2.2apnsize . . . . .	1054
8.1064.2.3auth . . . . .	1054
8.1064.2.4paddr . . . . .	1055
8.1064.2.5paddrv6 . . . . .	1055
8.1064.2.6name . . . . .	1055
8.1064.2.7namesize . . . . .	1055
8.1064.2.8ParamPresenceMask . . . . .	1055
8.1064.2.9pdptype . . . . .	1055
8.1064.2.10idns . . . . .	1055
8.1064.2.11idnsv6 . . . . .	1055

8.1064.2.12	ecdns	1055
8.1064.2.13	ecdnsv6	1055
8.1064.2.14	username	1055
8.1064.2.15	usersize	1055
8.1065	inpack_wds_GetDefaultProfileNum_t Struct Reference	1055
8.1065.1	Detailed Description	1055
8.1065.2	Field Documentation	1055
8.1065.2.1	index	1055
8.1065.2.2	ParamPresenceMask	1055
8.1066	inpack_wds_GetDefaultProfileV2_t Struct Reference	1055
8.1066.1	Detailed Description	1056
8.1066.2	Field Documentation	1058
8.1066.2.1	apnname	1058
8.1066.2.2	apnsize	1058
8.1066.2.3	auth	1058
8.1066.2.4	paddr	1058
8.1066.2.5	paddrv6	1058
8.1066.2.6	name	1058
8.1066.2.7	namesize	1058
8.1066.2.8	ParamPresenceMask	1058
8.1066.2.9	pdptype	1058
8.1066.2.10	idns	1058
8.1066.2.11	idnsv6	1058
8.1066.2.12	wd	1058
8.1066.2.13	wdsize	1058
8.1066.2.14	ecdns	1058
8.1066.2.15	ecdnsv6	1058
8.1066.2.16	username	1058
8.1066.2.17	usersize	1058
8.1067	inpack_wds_GetDormancyState_t Struct Reference	1058
8.1067.1	Detailed Description	1059
8.1067.2	Field Documentation	1059
8.1067.2.1	dormancyState	1059
8.1067.2.2	ParamPresenceMask	1059
8.1068	inpack_wds_GetLastMobileIPError_t Struct Reference	1059
8.1068.1	Detailed Description	1059
8.1068.2	Field Documentation	1059
8.1068.2.1	error	1059
8.1068.2.2	ParamPresenceMask	1059
8.1069	inpack_wds_GetMobileIP_t Struct Reference	1060

8.1069.1 Detailed Description	1060
8.1069.2 Field Documentation	1060
8.1069.2.1 mipMode	1060
8.1069.2.2 ParamPresenceMask	1060
8.1070.1 unpack_wds_GetMobileIPProfile_t Struct Reference	1060
8.1070.1 Detailed Description	1060
8.1070.2 Field Documentation	1062
8.1070.2.1 AAASPI	1062
8.1070.2.2 AAASState	1062
8.1070.2.3 address	1062
8.1070.2.4 enabled	1062
8.1070.2.5 HASPI	1062
8.1070.2.6 HASState	1062
8.1070.2.7 NAI	1062
8.1070.2.8 naiSize	1062
8.1070.2.9 ParamPresenceMask	1062
8.1070.2.10 primaryHA	1062
8.1070.2.11 revTunneling	1062
8.1070.2.12 secondaryHA	1062
8.1071.1 unpack_wds_GetPacketStatistics_t Struct Reference	1062
8.1071.1 Detailed Description	1063
8.1071.2 Field Documentation	1064
8.1071.2.1 ParamPresenceMask	1064
8.1071.2.2 pRXDroppedCount	1064
8.1071.2.3 pRXOkBytesCount	1064
8.1071.2.4 pRXOkBytesLastCall	1064
8.1071.2.5 pRXPacketErrors	1064
8.1071.2.6 pRXPacketOverflows	1064
8.1071.2.7 pRXPacketSuccesses	1064
8.1071.2.8 pTXDroppedCount	1064
8.1071.2.9 pTXOkBytesCount	1064
8.1071.2.10 pTXOkBytesLastCall	1064
8.1071.2.11 pTXPacketErrors	1064
8.1071.2.12 pTXPacketOverflows	1064
8.1071.2.13 pTXPacketSuccesses	1064
8.1072.1 unpack_wds_GetPacketStatus_t Struct Reference	1064
8.1072.1 Detailed Description	1065
8.1072.2 Field Documentation	1066
8.1072.2.1 ParamPresenceMask	1066
8.1072.2.2 pXDroppedCount	1066

8.1072.2.3XOkBytesCount . . . . .	1066
8.1072.2.4XOKBytesLastCall . . . . .	1066
8.1072.2.5XPacketErrors . . . . .	1066
8.1072.2.6XPacketOverflows . . . . .	1066
8.1072.2.7XPacketSuccesses . . . . .	1066
8.1072.2.8XDroppedCount . . . . .	1066
8.1072.2.9XOkBytesCount . . . . .	1066
8.1072.2.10XOKBytesLastCall . . . . .	1066
8.1072.2.11XPacketErrors . . . . .	1066
8.1072.2.12XPacketOverflows . . . . .	1066
8.1072.2.13XPacketSuccesses . . . . .	1066
8.1073inpack_wds_GetSessionDuration_t Struct Reference . . . . .	1066
8.1073.1Detailed Description . . . . .	1066
8.1073.2Field Documentation . . . . .	1067
8.1073.2.1callDuration . . . . .	1067
8.1073.2.2ParamPresenceMask . . . . .	1067
8.1074inpack_wds_GetSessionDurationV2_t Struct Reference . . . . .	1067
8.1074.1Detailed Description . . . . .	1067
8.1074.2Field Documentation . . . . .	1068
8.1074.2.1callDuration . . . . .	1068
8.1074.2.2ParamPresenceMask . . . . .	1068
8.1074.2.3pCallActiveDuration . . . . .	1068
8.1074.2.4pLastCallActiveDuration . . . . .	1068
8.1074.2.5pLastCallDuration . . . . .	1068
8.1075inpack_wds_GetSessionState_t Struct Reference . . . . .	1068
8.1075.1Detailed Description . . . . .	1068
8.1075.2Field Documentation . . . . .	1068
8.1075.2.1connectionStatus . . . . .	1068
8.1075.2.2ParamPresenceMask . . . . .	1068
8.1076inpack_wds_RMSetTransferStatistics_t Struct Reference . . . . .	1068
8.1076.1Detailed Description . . . . .	1068
8.1076.2Field Documentation . . . . .	1069
8.1076.2.1ParamPresenceMask . . . . .	1069
8.1077inpack_wds_SetMobileIPProfile_t Struct Reference . . . . .	1069
8.1077.1Detailed Description . . . . .	1069
8.1077.2Field Documentation . . . . .	1069
8.1077.2.1ParamPresenceMask . . . . .	1069
8.1078inpack_wds_SLQSCreateProfile_t Struct Reference . . . . .	1069
8.1078.1Detailed Description . . . . .	1069
8.1078.2Field Documentation . . . . .	1070



8.1078.2.1ParamPresenceMask . . . . .	1070
8.1078.2.2pCreateProfileOut . . . . .	1070
8.1078.2.3pProfileID . . . . .	1070
8.1078.2.4Tlvresult . . . . .	1070
8.1079anpack_wds_SLQSDDeleteProfile_t Struct Reference . . . . .	1070
8.1079.1Detailed Description . . . . .	1070
8.1079.2Field Documentation . . . . .	1070
8.1079.2.1extendedErrorCode . . . . .	1070
8.1079.2.2ParamPresenceMask . . . . .	1070
8.1080anpack_wds_SLQSDUNCallInfoCallBack_ind_t Struct Reference . . . . .	1070
8.1080.1Detailed Description . . . . .	1071
8.1080.2Field Documentation . . . . .	1072
8.1080.2.1CRTlv . . . . .	1072
8.1080.2.2CSTlv . . . . .	1072
8.1080.2.3DBTTlv . . . . .	1072
8.1080.2.4DSTlv . . . . .	1072
8.1080.2.5MCERTlv . . . . .	1072
8.1080.2.6ParamPresenceMask . . . . .	1072
8.1080.2.7RXBOTlv . . . . .	1072
8.1080.2.8TXBOTlv . . . . .	1072
8.1081anpack_wds_SLQSGet3GPPConfigItem_t Struct Reference . . . . .	1072
8.1081.1Detailed Description . . . . .	1072
8.1081.2Field Documentation . . . . .	1073
8.1081.2.1_3gppRelease . . . . .	1073
8.1081.2.2defaultPDNEnabled . . . . .	1073
8.1081.2.3TEAttachProfile . . . . .	1074
8.1081.2.4TEAttachProfileList . . . . .	1074
8.1081.2.5TEAttachProfileListLen . . . . .	1074
8.1081.2.6ParamPresenceMask . . . . .	1074
8.1081.2.7profileList . . . . .	1074
8.1082anpack_wds_SLQSGetCurrDataSystemStat_t Struct Reference . . . . .	1074
8.1082.1Detailed Description . . . . .	1074
8.1082.2Field Documentation . . . . .	1074
8.1082.2.1currNetworkInfo . . . . .	1074
8.1082.2.2networkInfoLen . . . . .	1074
8.1082.2.3ParamPresenceMask . . . . .	1075
8.1082.2.4prefNetwork . . . . .	1075
8.1083anpack_wds_SLQSGetCurrentChannelRate_t Struct Reference . . . . .	1075
8.1083.1Detailed Description . . . . .	1075
8.1083.2Field Documentation . . . . .	1076

8.1083.2.1current_channel_rx_rate . . . . .	1076
8.1083.2.2current_channel_tx_rate . . . . .	1076
8.1083.2.3max_channel_rx_rate . . . . .	1076
8.1083.2.4max_channel_tx_rate . . . . .	1076
8.1083.2.5ParamPresenceMask . . . . .	1076
8.1084inpack_wds_SLQSGetDataBearerTechnology_t Struct Reference . . . . .	1076
8.1084.1Detailed Description . . . . .	1076
8.1084.2Field Documentation . . . . .	1076
8.1084.2.1curDataBearerTechnology . . . . .	1076
8.1084.2.2dataBearerMask . . . . .	1077
8.1084.2.3astCallDataBearerTechnology . . . . .	1077
8.1084.2.4ParamPresenceMask . . . . .	1077
8.1085inpack_wds_SLQSGetDUNCallInfo_t Struct Reference . . . . .	1077
8.1085.1Detailed Description . . . . .	1077
8.1085.2Field Documentation . . . . .	1079
8.1085.2.1callEndReason . . . . .	1079
8.1085.2.2channelRate . . . . .	1079
8.1085.2.3connectionStatus . . . . .	1079
8.1085.2.4dataBearerTech . . . . .	1079
8.1085.2.5dormancyStatus . . . . .	1079
8.1085.2.6astCallDataBearerTech . . . . .	1079
8.1085.2.7astCallRXOKBytesCnt . . . . .	1079
8.1085.2.8astCallTXOKBytesCnt . . . . .	1080
8.1085.2.9mdmCallDurationActive . . . . .	1080
8.1085.2.10ParamPresenceMask . . . . .	1080
8.1085.2.11rxOKBytesCount . . . . .	1080
8.1085.2.12txOKBytesCount . . . . .	1080
8.1086inpack_wds_SLQSGetProfileSettings_t Struct Reference . . . . .	1080
8.1086.1Detailed Description . . . . .	1080
8.1086.2Field Documentation . . . . .	1080
8.1086.2.1ParamPresenceMask . . . . .	1080
8.1086.2.2pProfileSettings . . . . .	1080
8.1086.2.3ProfileType . . . . .	1080
8.1086.2.4Tlvresult . . . . .	1080
8.1087inpack_wds_SLQSGetProfileSettingsV2_t Struct Reference . . . . .	1080
8.1087.1Detailed Description . . . . .	1081
8.1087.2Field Documentation . . . . .	1081
8.1087.2.1ParamPresenceMask . . . . .	1081
8.1087.2.2pProfileSettings . . . . .	1081
8.1087.2.3ProfileType . . . . .	1081

8.1087.2.4Tlvresult	1081
8.1088npack_wds_SLQSGetRuntimeSettings_t Struct Reference	1081
8.1088.1Detailed Description	1082
8.1088.2Field Documentation	1083
8.1088.2.1APNName	1083
8.1088.2.2Authentication	1083
8.1088.2.3DomainList	1083
8.1088.2.4GPRSGrantedQoS	1083
8.1088.2.5GWAddressV4	1084
8.1088.2.6MCNflag	1084
8.1088.2.7PFfamilyPreference	1084
8.1088.2.8Pv4	1084
8.1088.2.9IPv6AddrInfo	1084
8.1088.2.10IPv6GWAddrInfo	1084
8.1088.2.11Mtu	1084
8.1088.2.12ParamPresenceMask	1084
8.1088.2.13CSCFAddrPCO	1084
8.1088.2.14CSCFFQDNAddrList	1084
8.1088.2.15DPTtype	1084
8.1088.2.16PrimaryDNSV4	1084
8.1088.2.17PrimaryDNSV6	1084
8.1088.2.18ProfileID	1084
8.1088.2.19ProfileName	1084
8.1088.2.20SecondaryDNSV4	1084
8.1088.2.21SecondaryDNSV6	1084
8.1088.2.22ServerAddrList	1084
8.1088.2.23SubnetMaskV4	1084
8.1088.2.24Technology	1084
8.1088.2.25MTSGrantedQoS	1084
8.1088.2.26Username	1084
8.1089npack_wds_SLQSModifyProfile_t Struct Reference	1084
8.1089.1Detailed Description	1084
8.1089.2Field Documentation	1085
8.1089.2.1ParamPresenceMask	1085
8.1089.2.2ExtErrorCode	1085
8.1090npack_wds_SLQSSetIPFamilyPreference_t Struct Reference	1085
8.1090.1Detailed Description	1085
8.1090.2Field Documentation	1085
8.1090.2.1ParamPresenceMask	1085
8.1090.2.2Tlvresult	1085

8.1091	<a href="#">inpack_wds_SLQSSetPacketSrvStatusCallback_t Struct Reference</a>	1085
8.1091.1	<a href="#">Detailed Description</a>	1086
8.1091.2	<a href="#">Field Documentation</a>	1087
8.1091.2.1	<a href="#">bearerID</a>	1087
8.1091.2.2	<a href="#">conn_status</a>	1087
8.1091.2.3	<a href="#">pFamily</a>	1087
8.1091.2.4	<a href="#">ParamPresenceMask</a>	1087
8.1091.2.5	<a href="#">reconfigReqd</a>	1087
8.1091.2.6	<a href="#">sessionEndReason</a>	1087
8.1091.2.7	<a href="#">techName</a>	1087
8.1091.2.8	<a href="#">verboseSessnEndReason</a>	1087
8.1091.2.9	<a href="#">verboseSessnEndReasonType</a>	1087
8.1092	<a href="#">inpack_wds_SLQSSetWdsEventCallback_ind_t Struct Reference</a>	1087
8.1092.1	<a href="#">Detailed Description</a>	1088
8.1092.2	<a href="#">Field Documentation</a>	1092
8.1092.2.1	<a href="#">currDBTechAvail</a>	1092
8.1092.2.2	<a href="#">currNWInfo</a>	1092
8.1092.2.3	<a href="#">dataSysStatAvail</a>	1092
8.1092.2.4	<a href="#">dBTechAvail</a>	1092
8.1092.2.5	<a href="#">dBtechExtAvail</a>	1093
8.1092.2.6	<a href="#">dBTechExtRatValue</a>	1093
8.1092.2.7	<a href="#">dBTechExtSoMask</a>	1093
8.1092.2.8	<a href="#">dBTechnology</a>	1093
8.1092.2.9	<a href="#">dBtechnologyExt</a>	1093
8.1092.2.10	<a href="#">dormancyStatAvail</a>	1093
8.1092.2.11	<a href="#">dormancyStatus</a>	1093
8.1092.2.12	<a href="#">ipstatAvail</a>	1093
8.1092.2.13	<a href="#">ipStatus</a>	1093
8.1092.2.14	<a href="#">netInfoLen</a>	1093
8.1092.2.15	<a href="#">ParamPresenceMask</a>	1093
8.1092.2.16	<a href="#">refNetwork</a>	1093
8.1092.2.17	<a href="#">ratMask</a>	1093
8.1092.2.18	<a href="#">_bytes</a>	1093
8.1092.2.19	<a href="#">_pkts</a>	1093
8.1092.2.20	<a href="#">Mask</a>	1093
8.1092.2.21	<a href="#">_bytes</a>	1093
8.1092.2.22	<a href="#">_pkts</a>	1093
8.1092.2.23	<a href="#">erStatAvail</a>	1093
8.1093	<a href="#">inpack_wds_SLQSSetDHCPv4ClientConfig_t Struct Reference</a>	1093
8.1093.1	<a href="#">Detailed Description</a>	1093

8.1093.2Field Documentation	1094
8.1093.2.1ParamPresenceMask	1094
8.1093.2.2pHwConfig	1094
8.1093.2.3pRequestOptionList	1094
8.1094.1unpack_wds_SLQSSGetLoopback_t Struct Reference	1094
8.1094.1Detailed Description	1094
8.1094.2Field Documentation	1095
8.1094.2.1ByteLoopbackMode	1095
8.1094.2.2ByteLoopbackMultiplier	1095
8.1094.2.3ParamPresenceMask	1095
8.1095.1unpack_wds_SLQSSStartDataSession_t Struct Reference	1095
8.1095.1Detailed Description	1095
8.1095.2Field Documentation	1095
8.1095.2.1ParamPresenceMask	1095
8.1095.2.2pFailureReason	1095
8.1095.2.3psid	1096
8.1095.2.4pVerboseFailReasonType	1096
8.1095.2.5pVerboseFailureReason	1096
8.1096.1unpack_wds_SLQSSwiProfileChangeCallback_Ind_t Struct Reference	1096
8.1096.1Detailed Description	1096
8.1096.2Field Documentation	1096
8.1096.2.1ParamPresenceMask	1096
8.1096.2.2ProfileTlv	1096
8.1096.2.3srcTlv	1096
8.1097.1unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t Struct Reference	1096
8.1097.1Detailed Description	1097
8.1097.2Field Documentation	1098
8.1097.2.1apnName	1098
8.1097.2.2bearerId	1098
8.1097.2.3contextId	1098
8.1097.2.4pv4Address	1098
8.1097.2.5pv4GWAddress	1098
8.1097.2.6pv6Address	1098
8.1097.2.7pv6GWAddress	1098
8.1097.2.8ParamPresenceMask	1098
8.1097.2.9prDNSIPv4Address	1098
8.1097.2.10prDNSIPv6Address	1099
8.1097.2.11prPCSCFIPv4Address	1099
8.1097.2.12prPCSCFIPv6Address	1099
8.1097.2.13seDNSIPv4Address	1099

8.1097.2.1 <del>4</del> <del>5</del> PCSCFIPv6Address	1099
8.1097.2.1 <del>5</del> <del>6</del> PCSCFIPv4Address	1099
8.1097.2.1 <del>6</del> <del>7</del> PCSCFIPv6Address	1099
8.1098 <del>8</del> UnPackGetProfileSettingOut Struct Reference	1099
8.1098.1Detailed Description	1099
8.1098.2Field Documentation	1099
8.1098.2.1curProfile	1099
8.1098.2.2pExtErrCode	1099
8.1099 <del>9</del> UnPackGetProfileSettingOutV2 Struct Reference	1099
8.1099.1Detailed Description	1100
8.1099.2Field Documentation	1100
8.1099.2.1curProfile	1100
8.1099.2.2pExtErrCode	1100
8.1100 <del>0</del> UnpackSwiAvmsEventReportBinaryUpdateSessionInfo Struct Reference	1100
8.1100.1Detailed Description	1100
8.1100.2Field Documentation	1101
8.1100.2.1bBinaryType	1101
8.1100.2.2bSerity	1101
8.1100.2.3bState	1101
8.1100.2.4bUserInputRequest	1101
8.1100.2.5szDescription	1101
8.1100.2.6szName	1102
8.1100.2.7szVersion	1102
8.1100.2.8TlvPresent	1102
8.1100.2.9ulPkgDownloadComplete	1102
8.1100.2.10ulPkgDownloadSize	1102
8.1100.2.11wDescriptionLength	1102
8.1100.2.12wNameLength	1102
8.1100.2.13wUpdateCompleteStatus	1102
8.1100.2.14wUserInputTimeout	1102
8.1100.2.15wVersionLength	1102
8.1100UnpackSwiAvmsEventReportConfig Struct Reference	1102
8.1101.1Detailed Description	1102
8.1101.2Field Documentation	1103
8.1101.2.1bState	1103
8.1101.2.2bUserInputRequest	1103
8.1101.2.3szAlertMsg	1103
8.1101.2.4TlvPresent	1103
8.1101.2.5wAlertMsgLength	1103
8.1101.2.6wUserInputTimeout	1103

8.1102	InpackSwiAvmsEventReportConnectionRequest Struct Reference	1103
8.1102.1	Detailed Description	1103
8.1102.2	Field Documentation	1103
8.1102.2.1	bUserInputRequest	1103
8.1102.2.2	TlvPresent	1103
8.1102.2.3	wUserInputTimeout	1103
8.1103	InpackSwiAvmsEventReportDataSessionStatus Struct Reference	1103
8.1103.1	Detailed Description	1104
8.1103.2	Field Documentation	1104
8.1103.2.1	bType	1104
8.1103.2.2	TlvPresent	1104
8.1103.2.3	wErrorCode	1104
8.1104	InpackSwiAvmsEventReportHTTPStatus Struct Reference	1104
8.1104.1	Detailed Description	1104
8.1104.2	Field Documentation	1105
8.1104.2.1	TlvPresent	1105
8.1104.2.2	wHTTPStatus	1106
8.1105	InpackSwiAvmsEventReportNotification Struct Reference	1106
8.1105.1	Detailed Description	1106
8.1105.2	Field Documentation	1106
8.1105.2.1	bNotification	1107
8.1105.2.2	TlvPresent	1107
8.1105.2.3	wSessionStatus	1107
8.1106	InpackSwiAvmsEventReportPackageID Struct Reference	1107
8.1106.1	Detailed Description	1107
8.1106.2	Field Documentation	1107
8.1106.2.1	bPackageID	1107
8.1106.2.2	TlvPresent	1107
8.1107	InpackSwiAvmsEventReportRegStatus Struct Reference	1107
8.1107.1	Detailed Description	1107
8.1107.2	Field Documentation	1107
8.1107.2.1	bRegStatus	1107
8.1107.2.2	TlvPresent	1108
8.1108	InpackSwiAvmsEventReportSessionType Struct Reference	1108
8.1108.1	Detailed Description	1108
8.1108.2	Field Documentation	1108
8.1108.2.1	bType	1108
8.1108.2.2	TlvPresent	1108
8.1109	InpackSwiAvmsEventReportWAMSPParamChange Struct Reference	1108
8.1109.1	Detailed Description	1108

8.1109.2Field Documentation	1109
8.1109.2.1TlvPresent	1109
8.1109.2.2wWamsChangeMask	1109
8.1110unpackWdsProfileParam Union Reference	1109
8.1110.1Detailed Description	1109
8.1110.2Field Documentation	1109
8.1110.2.1SlqsProfile3GPP	1109
8.1110.2.2SlqsProfile3GPP2	1109
8.1111unpackWdsProfileParamV2 Union Reference	1109
8.1111.1Detailed Description	1109
8.1111.2Field Documentation	1110
8.1111.2.1SlqsProfile3GPP	1110
8.1111.2.2SlqsProfile3GPP2	1110
8.1112voice_airTimer Struct Reference	1110
8.1112.1Detailed Description	1110
8.1112.2Field Documentation	1110
8.1112.2.1airTimerValue	1110
8.1112.2.2hamID	1110
8.1113voice_allCallsAlphaIDInfo Struct Reference	1110
8.1113.1Detailed Description	1110
8.1113.2Field Documentation	1111
8.1113.2.1AlphaIDInfo	1111
8.1113.2.2callID	1111
8.1114voice_allCallsDiagInfo Struct Reference	1111
8.1114.1Detailed Description	1111
8.1114.2Field Documentation	1111
8.1114.2.1callID	1111
8.1114.2.2DiagInfo	1111
8.1115voice_allCallsUUSInfo Struct Reference	1111
8.1115.1Detailed Description	1111
8.1115.2Field Documentation	1112
8.1115.2.1callID	1112
8.1115.2.2uusInfo	1112
8.1116voice_alphaIDInfo Struct Reference	1112
8.1116.1Detailed Description	1112
8.1116.2Field Documentation	1112
8.1116.2.1alphaDcs	1112
8.1116.2.2alphaLen	1112
8.1116.2.3alphaText	1112
8.1117voice_arrAlertingPattern Struct Reference	1112



8.1117.1Detailed Description . . . . .	1113
8.1117.2Field Documentation . . . . .	1113
8.1117.2.1alertingPattern . . . . .	1113
8.1117.2.2callID . . . . .	1113
8.1117.2.3numInstances . . . . .	1113
8.1118voice_arrAlertingType Struct Reference . . . . .	1113
8.1118.1Detailed Description . . . . .	1113
8.1118.2Field Documentation . . . . .	1114
8.1118.2.1AlertingType . . . . .	1114
8.1118.2.2callID . . . . .	1114
8.1118.2.3numInstances . . . . .	1114
8.1119voice_arrAlphaID Struct Reference . . . . .	1114
8.1119.1Detailed Description . . . . .	1114
8.1119.2Field Documentation . . . . .	1114
8.1119.2.1allCallsAlphaIDInfoArr . . . . .	1114
8.1119.2.2numInstances . . . . .	1114
8.1120voice_arrCalledPartyNum Struct Reference . . . . .	1114
8.1120.1Detailed Description . . . . .	1115
8.1120.2Field Documentation . . . . .	1115
8.1120.2.1CalledPartyNum . . . . .	1115
8.1120.2.2numInstances . . . . .	1115
8.1121voice_arrCallEndReason Struct Reference . . . . .	1115
8.1121.1Detailed Description . . . . .	1115
8.1121.2Field Documentation . . . . .	1116
8.1121.2.1callEndReason . . . . .	1116
8.1121.2.2callID . . . . .	1116
8.1121.2.3numInstances . . . . .	1116
8.1122voice_arrCallInfo Struct Reference . . . . .	1116
8.1122.1Detailed Description . . . . .	1116
8.1122.2Field Documentation . . . . .	1116
8.1122.2.1getAllCallInfo . . . . .	1116
8.1122.2.2numInstances . . . . .	1116
8.1123voice_arrConnectPartyNum Struct Reference . . . . .	1116
8.1123.1Detailed Description . . . . .	1116
8.1123.2Field Documentation . . . . .	1117
8.1123.2.1ConnectedPartyNum . . . . .	1117
8.1123.2.2numInstances . . . . .	1117
8.1124voice_arrDiagInfo Struct Reference . . . . .	1117
8.1124.1Detailed Description . . . . .	1117
8.1124.2Field Documentation . . . . .	1117

8.1124.2.1DiagInfo . . . . .	1117
8.1124.2.2numInstances . . . . .	1117
8.1125voice_arrRedirPartyNum Struct Reference . . . . .	1117
8.1125.1Detailed Description . . . . .	1117
8.1125.2Field Documentation . . . . .	1118
8.1125.2.1numInstances . . . . .	1118
8.1125.2.2RedirPartyNum . . . . .	1118
8.1126voice_arrRemotePartyName Struct Reference . . . . .	1118
8.1126.1Detailed Description . . . . .	1118
8.1126.2Field Documentation . . . . .	1118
8.1126.2.1GetAllCallRmtPtyName . . . . .	1118
8.1126.2.2numInstances . . . . .	1118
8.1127voice_arrRemotePartyNum Struct Reference . . . . .	1118
8.1127.1Detailed Description . . . . .	1119
8.1127.2Field Documentation . . . . .	1119
8.1127.2.1numInstances . . . . .	1119
8.1127.2.2RmtPtyNum . . . . .	1119
8.1128voice_arrSvcOption Struct Reference . . . . .	1119
8.1128.1Detailed Description . . . . .	1119
8.1128.2Field Documentation . . . . .	1119
8.1128.2.1callID . . . . .	1119
8.1128.2.2numInstances . . . . .	1119
8.1128.2.3srvOption . . . . .	1120
8.1129voice_arrUUSInfo Struct Reference . . . . .	1120
8.1129.1Detailed Description . . . . .	1120
8.1129.2Field Documentation . . . . .	1120
8.1129.2.1AllCallsUUSInfo . . . . .	1120
8.1129.2.2numInstances . . . . .	1120
8.1130voice_burstDTMFInfo Struct Reference . . . . .	1120
8.1130.1Detailed Description . . . . .	1120
8.1130.2Field Documentation . . . . .	1121
8.1130.2.1digitCnt . . . . .	1121
8.1130.2.2pCallID . . . . .	1121
8.1130.2.3pDigitBuff . . . . .	1121
8.1131voice_calledPartyInfo Struct Reference . . . . .	1121
8.1131.1Detailed Description . . . . .	1121
8.1131.2Field Documentation . . . . .	1122
8.1131.2.1number . . . . .	1122
8.1131.2.2numLen . . . . .	1122
8.1131.2.3numPlan . . . . .	1122

8.1131.2.4numType . . . . .	1122
8.1131.2.5PI . . . . .	1122
8.1131.2.6SI . . . . .	1122
8.1132oice_calledPartySubAdd Struct Reference . . . . .	1122
8.1132.1Detailed Description . . . . .	1123
8.1132.2Field Documentation . . . . .	1123
8.1132.2.1textBit . . . . .	1123
8.1132.2.2oddEvenInd . . . . .	1123
8.1132.2.3subAddr . . . . .	1123
8.1132.2.4subAddrLen . . . . .	1123
8.1132.2.5subAddrType . . . . .	1123
8.1133oice_callerIDInfo Struct Reference . . . . .	1123
8.1133.1Detailed Description . . . . .	1124
8.1133.2Field Documentation . . . . .	1124
8.1133.2.1callerID . . . . .	1124
8.1133.2.2callerIDLen . . . . .	1124
8.1133.2.3PI . . . . .	1124
8.1134oice_callFwdTypeAndPlan Struct Reference . . . . .	1124
8.1134.1Detailed Description . . . . .	1124
8.1134.2Field Documentation . . . . .	1125
8.1134.2.1numberPlan . . . . .	1125
8.1134.2.2numberType . . . . .	1125
8.1135oice_callFWExtInfo Struct Reference . . . . .	1125
8.1135.1Detailed Description . . . . .	1125
8.1135.2Field Documentation . . . . .	1126
8.1135.2.1noReplyTimer . . . . .	1126
8.1135.2.2number . . . . .	1126
8.1135.2.3numLen . . . . .	1127
8.1135.2.4numPlan . . . . .	1127
8.1135.2.5numType . . . . .	1127
8.1135.2.6PI . . . . .	1127
8.1135.2.7SI . . . . .	1127
8.1135.2.8SvcClass . . . . .	1127
8.1135.2.9SvcStatus . . . . .	1127
8.1136oice_callFWInfo Struct Reference . . . . .	1127
8.1136.1Detailed Description . . . . .	1127
8.1136.2Field Documentation . . . . .	1127
8.1136.2.1noReplyTimer . . . . .	1127
8.1136.2.2number . . . . .	1128
8.1136.2.3numLen . . . . .	1128

8.1136.2.4SvcClass . . . . .	1128
8.1136.2.5SvcStatus . . . . .	1128
8.1137voice_callInfo Struct Reference . . . . .	1128
8.1137.1Detailed Description . . . . .	1128
8.1137.2Field Documentation . . . . .	1129
8.1137.2.1callID . . . . .	1129
8.1137.2.2callState . . . . .	1129
8.1137.2.3callType . . . . .	1129
8.1137.2.4direction . . . . .	1129
8.1137.2.5mode . . . . .	1129
8.1138voice_callingPartyInfo Struct Reference . . . . .	1129
8.1138.1Detailed Description . . . . .	1130
8.1138.2Field Documentation . . . . .	1131
8.1138.2.1number . . . . .	1131
8.1138.2.2numLen . . . . .	1131
8.1138.2.3numPlan . . . . .	1131
8.1138.2.4numType . . . . .	1131
8.1138.2.5PI . . . . .	1131
8.1138.2.6SI . . . . .	1131
8.1139voice_ccSUPSType Struct Reference . . . . .	1131
8.1139.1Detailed Description . . . . .	1131
8.1139.2Field Documentation . . . . .	1131
8.1139.2.1reason . . . . .	1132
8.1139.2.2svcType . . . . .	1132
8.1140voice_CLIPResp Struct Reference . . . . .	1132
8.1140.1Detailed Description . . . . .	1132
8.1140.2Field Documentation . . . . .	1132
8.1140.2.1ActiveStatus . . . . .	1132
8.1140.2.2ProvisionStatus . . . . .	1132
8.1141voice_CLIRResp Struct Reference . . . . .	1132
8.1141.1Detailed Description . . . . .	1132
8.1141.2Field Documentation . . . . .	1133
8.1141.2.1ActiveStatus . . . . .	1133
8.1141.2.2ProvisionStatus . . . . .	1133
8.1142voice_CNAPResp Struct Reference . . . . .	1133
8.1142.1Detailed Description . . . . .	1133
8.1142.2Field Documentation . . . . .	1134
8.1142.2.1ActiveStatus . . . . .	1134
8.1142.2.2ProvisionStatus . . . . .	1134
8.1143voice_COLPResp Struct Reference . . . . .	1134

8.1143.1Detailed Description . . . . .	1134
8.1143.2Field Documentation . . . . .	1134
8.1143.2.1ActiveStatus . . . . .	1134
8.1143.2.2ProvisionStatus . . . . .	1134
8.1144voice_COLRResp Struct Reference . . . . .	1134
8.1144.1Detailed Description . . . . .	1134
8.1144.2Field Documentation . . . . .	1135
8.1144.2.1ActiveStatus . . . . .	1135
8.1144.2.2ProvisionStatus . . . . .	1135
8.1145voice_connectNumInfo Struct Reference . . . . .	1135
8.1145.1Detailed Description . . . . .	1135
8.1145.2Field Documentation . . . . .	1136
8.1145.2.1callerID . . . . .	1136
8.1145.2.2callerIDLen . . . . .	1136
8.1145.2.3numPlan . . . . .	1136
8.1145.2.4numPresInd . . . . .	1137
8.1145.2.5numType . . . . .	1137
8.1145.2.6screeningInd . . . . .	1137
8.1146voice_CUGInfo Struct Reference . . . . .	1137
8.1146.1Detailed Description . . . . .	1137
8.1146.2Field Documentation . . . . .	1137
8.1146.2.1CUGIndex . . . . .	1137
8.1146.2.2SuppOA . . . . .	1137
8.1146.2.3SuppPrefCUG . . . . .	1137
8.1147voice_curAMRConfig Struct Reference . . . . .	1137
8.1147.1Detailed Description . . . . .	1137
8.1147.2Field Documentation . . . . .	1138
8.1147.2.1gsmAmrStat . . . . .	1138
8.1147.2.2wcdmaAmrStat . . . . .	1138
8.1148voice_diagInfo Struct Reference . . . . .	1138
8.1148.1Detailed Description . . . . .	1138
8.1148.2Field Documentation . . . . .	1138
8.1148.2.1diagInfoLen . . . . .	1138
8.1148.2.2diagnosticInfo . . . . .	1138
8.1149voice_DTMFInfo Struct Reference . . . . .	1139
8.1149.1Detailed Description . . . . .	1139
8.1149.2Field Documentation . . . . .	1139
8.1149.2.1callID . . . . .	1139
8.1149.2.2digitBuff . . . . .	1139
8.1149.2.3digitCnt . . . . .	1139

8.1149.2.4DTMFEvent . . . . .	1139
8.1150voice_DTMFLengths Struct Reference . . . . .	1139
8.1150.1Detailed Description . . . . .	1140
8.1150.2Field Documentation . . . . .	1140
8.1150.2.1DTMFInterdigitInterval . . . . .	1140
8.1150.2.2DTMFPulseWidth . . . . .	1140
8.1151voice_ECTNum Struct Reference . . . . .	1140
8.1151.1Detailed Description . . . . .	1140
8.1151.2Field Documentation . . . . .	1141
8.1151.2.1ECTCallState . . . . .	1141
8.1151.2.2number . . . . .	1141
8.1151.2.3presentationInd . . . . .	1141
8.1152voice_extDispRecInfo Struct Reference . . . . .	1141
8.1152.1Detailed Description . . . . .	1141
8.1152.2Field Documentation . . . . .	1141
8.1152.2.1dispType . . . . .	1141
8.1152.2.2extDispInfo . . . . .	1141
8.1152.2.3extDispInfoLen . . . . .	1142
8.1153voice_getAllCallInformation Struct Reference . . . . .	1142
8.1153.1Detailed Description . . . . .	1142
8.1153.2Field Documentation . . . . .	1142
8.1153.2.1ALS . . . . .	1142
8.1153.2.2Callinfo . . . . .	1142
8.1153.2.3sEmpty . . . . .	1142
8.1154voice_getAllCallRmtPtyName Struct Reference . . . . .	1142
8.1154.1Detailed Description . . . . .	1142
8.1154.2Field Documentation . . . . .	1143
8.1154.2.1callID . . . . .	1143
8.1154.2.2RemotePartyName . . . . .	1143
8.1155voice_getAllCallRmtPtyNum Struct Reference . . . . .	1143
8.1155.1Detailed Description . . . . .	1143
8.1155.2Field Documentation . . . . .	1143
8.1155.2.1callID . . . . .	1143
8.1155.2.2RemotePartyNum . . . . .	1143
8.1156voice_getCallFWExtInfo Struct Reference . . . . .	1143
8.1156.1Detailed Description . . . . .	1143
8.1156.2Field Documentation . . . . .	1144
8.1156.2.1CallFWExtInfo . . . . .	1144
8.1156.2.2numInstances . . . . .	1144
8.1157voice_getCallFWInfo Struct Reference . . . . .	1144

8.1157.1Detailed Description	1144
8.1157.2Field Documentation	1144
8.1157.2.1CallFWInfo	1144
8.1157.2.2numInstances	1144
8.1158voice_lineCtrlInfo Struct Reference	1144
8.1158.1Detailed Description	1145
8.1158.2Field Documentation	1145
8.1158.2.1polarityIncluded	1145
8.1158.2.2pwrDenialTime	1145
8.1158.2.3revPolarity	1145
8.1158.2.4toggleMode	1145
8.1159voice_newPwdData Struct Reference	1145
8.1159.1Detailed Description	1145
8.1159.2Field Documentation	1146
8.1159.2.1newPwd	1146
8.1159.2.2newPwdAgain	1146
8.1160voice_NSSAudioCtrl Struct Reference	1146
8.1160.1Detailed Description	1146
8.1160.2Field Documentation	1146
8.1160.2.1downLink	1146
8.1160.2.2upLink	1146
8.1161voice_peerNumberInfo Struct Reference	1146
8.1161.1Detailed Description	1146
8.1161.2Field Documentation	1147
8.1161.2.1callID	1148
8.1161.2.2number	1148
8.1161.2.3numLen	1148
8.1161.2.4numPI	1148
8.1161.2.5numPlan	1148
8.1161.2.6numSI	1148
8.1161.2.7numType	1148
8.1162voice_prefVoiceSO Struct Reference	1148
8.1162.1Detailed Description	1148
8.1162.2Field Documentation	1150
8.1162.2.1evrcCapability	1150
8.1162.2.2homeOrigVoiceSO	1150
8.1162.2.3homePageVoiceSO	1150
8.1162.2.4namID	1150
8.1162.2.5oamOrigVoiceSO	1150
8.1163voice_redirNumInfo Struct Reference	1150

8.1163.1 Detailed Description	1150
8.1163.2 Field Documentation	1151
8.1163.2.1 number	1151
8.1163.2.2 numLen	1151
8.1163.2.3 numPlan	1151
8.1163.2.4 numType	1151
8.1163.2.5 PI	1151
8.1163.2.6 reason	1151
8.1163.2.7 SI	1151
8.1164 voice_remotePartyName Struct Reference	1152
8.1164.1 Detailed Description	1152
8.1164.2 Field Documentation	1152
8.1164.2.1 callerName	1152
8.1164.2.2 codingScheme	1152
8.1164.2.3 nameLen	1152
8.1164.2.4 namePI	1152
8.1165 voice_remotePartyNum Struct Reference	1152
8.1165.1 Detailed Description	1153
8.1165.2 Field Documentation	1153
8.1165.2.1 numLen	1153
8.1165.2.2 presentationInd	1153
8.1165.2.3 remPartyNumber	1153
8.1166 voice_roamTimer Struct Reference	1153
8.1166.1 Detailed Description	1153
8.1166.2 Field Documentation	1154
8.1166.2.1 namID	1154
8.1166.2.2 roamTimerValue	1154
8.1167 voice_signalInfo Struct Reference	1154
8.1167.1 Detailed Description	1154
8.1167.2 Field Documentation	1154
8.1167.2.1 alertPitch	1154
8.1167.2.2 signal	1154
8.1167.2.3 signalType	1154
8.1168 voice_SUPSInfo Struct Reference	1155
8.1168.1 Detailed Description	1155
8.1168.2 Field Documentation	1155
8.1168.2.1 isModByCC	1155
8.1168.2.2 svcType	1155
8.1169 voice_USSDNotificationNetworkInfo Struct Reference	1155
8.1169.1 Detailed Description	1155



8.1169.2Field Documentation . . . . .	1156
8.1169.2.1networkInfo . . . . .	1156
8.1169.2.2lvPresent . . . . .	1156
8.1170voice_USSInfo Struct Reference . . . . .	1156
8.1170.1Detailed Description . . . . .	1156
8.1170.2Field Documentation . . . . .	1156
8.1170.2.1ussData . . . . .	1156
8.1170.2.2ussDCS . . . . .	1156
8.1170.2.3ussLen . . . . .	1156
8.1171voice_UUSInfo Struct Reference . . . . .	1156
8.1171.1Detailed Description . . . . .	1157
8.1171.2Field Documentation . . . . .	1157
8.1171.2.1UUSData . . . . .	1157
8.1171.2.2JUUSDataLen . . . . .	1157
8.1171.2.3JUUSDCS . . . . .	1157
8.1171.2.4JUUSType . . . . .	1157
8.1172nds_channelRate Struct Reference . . . . .	1158
8.1172.1Detailed Description . . . . .	1158
8.1172.2Field Documentation . . . . .	1158
8.1172.2.1CurrChanRxRate . . . . .	1158
8.1172.2.2CurrChanTxRate . . . . .	1158
8.1173nds_ChannelRateTlv Struct Reference . . . . .	1158
8.1173.1Detailed Description . . . . .	1158
8.1173.2Field Documentation . . . . .	1158
8.1173.2.1ChannelRate . . . . .	1158
8.1173.2.2TlvPresent . . . . .	1158
8.1174nds_ConnStatusTlv Struct Reference . . . . .	1159
8.1174.1Detailed Description . . . . .	1159
8.1174.2Field Documentation . . . . .	1159
8.1174.2.1MDMConnStatus . . . . .	1159
8.1174.2.2TlvPresent . . . . .	1159
8.1175nds_currNetworkInfo Struct Reference . . . . .	1159
8.1175.1Detailed Description . . . . .	1159
8.1175.2Field Documentation . . . . .	1160
8.1175.2.1NetworkType . . . . .	1160
8.1175.2.2RATMask . . . . .	1160
8.1175.2.3SOMask . . . . .	1160
8.1176nds_DataBearTechTlv Struct Reference . . . . .	1160
8.1176.1Detailed Description . . . . .	1160
8.1176.2Field Documentation . . . . .	1161

8.1176.2.1DataBearerTech . . . . .	1161
8.1176.2.2TlvPresent . . . . .	1161
8.1177wds_DataULongLongTlv Struct Reference . . . . .	1161
8.1177.1Detailed Description . . . . .	1161
8.1177.2Field Documentation . . . . .	1162
8.1177.2.1TlvPresent . . . . .	1162
8.1177.2.2AllData . . . . .	1162
8.1178wds_DataULongTlv Struct Reference . . . . .	1162
8.1178.1Detailed Description . . . . .	1162
8.1178.2Field Documentation . . . . .	1162
8.1178.2.1TlvPresent . . . . .	1162
8.1178.2.2AllData . . . . .	1162
8.1179wds_DHCPLeaseOptTlv Struct Reference . . . . .	1162
8.1179.1Detailed Description . . . . .	1162
8.1179.2Field Documentation . . . . .	1162
8.1179.2.1numOpt . . . . .	1162
8.1179.2.2optList . . . . .	1163
8.1179.2.3optListData . . . . .	1163
8.1179.2.4TlvPresent . . . . .	1163
8.1180wds_DHCPLeaseStateTlv Struct Reference . . . . .	1163
8.1180.1Detailed Description . . . . .	1163
8.1180.2Field Documentation . . . . .	1163
8.1180.2.1leaseState . . . . .	1163
8.1180.2.2TlvPresent . . . . .	1163
8.1181wds_DHCPOpt Struct Reference . . . . .	1163
8.1181.1Detailed Description . . . . .	1163
8.1181.2Field Documentation . . . . .	1164
8.1181.2.1optCode . . . . .	1164
8.1181.2.2optValLen . . . . .	1164
8.1181.2.3optVal . . . . .	1164
8.1182wds_DHCPProfiledTlv Struct Reference . . . . .	1164
8.1182.1Detailed Description . . . . .	1164
8.1182.2Field Documentation . . . . .	1164
8.1182.2.1profileId . . . . .	1164
8.1182.2.2profileType . . . . .	1164
8.1182.2.3TlvPresent . . . . .	1164
8.1183wds_DHCPv4HWConfig Struct Reference . . . . .	1164
8.1183.1Detailed Description . . . . .	1165
8.1183.2Field Documentation . . . . .	1165
8.1183.2.1chaddr . . . . .	1165

8.1183.2.2chaddrLen . . . . .	1165
8.1183.2.3hwType . . . . .	1165
8.1184.1nds_DHCPv4Option Struct Reference . . . . .	1165
8.1184.1.1Detailed Description . . . . .	1165
8.1184.2Field Documentation . . . . .	1166
8.1184.2.1optCode . . . . .	1166
8.1184.2.2optVal . . . . .	1166
8.1184.2.3optValLen . . . . .	1166
8.1185.1nds_DHCPv4OptionList Struct Reference . . . . .	1166
8.1185.1.1Detailed Description . . . . .	1166
8.1185.2Field Documentation . . . . .	1166
8.1185.2.1numOpt . . . . .	1166
8.1185.2.2pOptList . . . . .	1166
8.1186.1nds_DHCPv4Profile Struct Reference . . . . .	1166
8.1186.1.1Detailed Description . . . . .	1166
8.1186.2Field Documentation . . . . .	1167
8.1186.2.1profileId . . . . .	1167
8.1186.2.2profileType . . . . .	1167
8.1187.1nds_Domain Struct Reference . . . . .	1167
8.1187.1.1Detailed Description . . . . .	1167
8.1187.2Field Documentation . . . . .	1167
8.1187.2.1domainLen . . . . .	1167
8.1187.2.2domainName . . . . .	1167
8.1188.1nds_DomainNameList Struct Reference . . . . .	1167
8.1188.1.1Detailed Description . . . . .	1167
8.1188.2Field Documentation . . . . .	1168
8.1188.2.1domain . . . . .	1168
8.1188.2.2numInstances . . . . .	1168
8.1189.1nds_DormStatTlv Struct Reference . . . . .	1168
8.1189.1.1Detailed Description . . . . .	1168
8.1189.2Field Documentation . . . . .	1168
8.1189.2.1DormancyStat . . . . .	1168
8.1189.2.2TlvPresent . . . . .	1168
8.1190.1nds_GPRSQoS Struct Reference . . . . .	1168
8.1190.1.1Detailed Description . . . . .	1169
8.1190.2Field Documentation . . . . .	1169
8.1190.2.1delayClass . . . . .	1169
8.1190.2.2meanThroughputClass . . . . .	1169
8.1190.2.3peakThroughputClass . . . . .	1169
8.1190.2.4precedenceClass . . . . .	1169

8.1190.2.5reliabilityClass . . . . .	1169
8.1191nds_IPv4AdTlv Struct Reference . . . . .	1169
8.1191.1Detailed Description . . . . .	1169
8.1191.2Field Documentation . . . . .	1170
8.1191.2.1IPv4Addr . . . . .	1170
8.1191.2.2TlvPresent . . . . .	1170
8.1192nds_IPV6AddressInfo Struct Reference . . . . .	1170
8.1192.1Detailed Description . . . . .	1170
8.1192.2Field Documentation . . . . .	1170
8.1192.2.1IPAddressV6 . . . . .	1170
8.1192.2.2IPv6PrefixLen . . . . .	1170
8.1193nds_IPV6GWAddressInfo Struct Reference . . . . .	1170
8.1193.1Detailed Description . . . . .	1170
8.1193.2Field Documentation . . . . .	1171
8.1193.2.1gwAddressV6 . . . . .	1171
8.1193.2.2gwV6PrefixLen . . . . .	1171
8.1194nds_LastMdmCallEndRsnTlv Struct Reference . . . . .	1171
8.1194.1Detailed Description . . . . .	1171
8.1194.2Field Documentation . . . . .	1171
8.1194.2.1CallEndReason . . . . .	1171
8.1194.2.2TlvPresent . . . . .	1171
8.1195nds_PCSCFFQDNAddress Struct Reference . . . . .	1171
8.1195.1Detailed Description . . . . .	1172
8.1195.2Field Documentation . . . . .	1172
8.1195.2.1fqdnAddr . . . . .	1172
8.1195.2.2fqdnLen . . . . .	1172
8.1196nds_PCSCFFQDNAddressList Struct Reference . . . . .	1172
8.1196.1Detailed Description . . . . .	1172
8.1196.2Field Documentation . . . . .	1172
8.1196.2.1numInstances . . . . .	1172
8.1196.2.2pcsfQDNAddress . . . . .	1172
8.1197nds_PCSCFIPv4ServerAddressList Struct Reference . . . . .	1172
8.1197.1Detailed Description . . . . .	1173
8.1197.2Field Documentation . . . . .	1173
8.1197.2.1numInstances . . . . .	1173
8.1197.2.2pscfIPv4Addr . . . . .	1173
8.1198nds_profileChange Struct Reference . . . . .	1173
8.1198.1Detailed Description . . . . .	1173
8.1198.2Field Documentation . . . . .	1173
8.1198.2.1profileIdx . . . . .	1173

8.1198.2.2profileType . . . . .	1173
8.1199.1ds_ProfileIdentifier Struct Reference . . . . .	1173
8.1199.1Detailed Description . . . . .	1174
8.1199.2Field Documentation . . . . .	1174
8.1199.2.1profileIndex . . . . .	1174
8.1199.2.2profileType . . . . .	1174
8.1200.1ds_profileInfo Union Reference . . . . .	1174
8.1200.1Detailed Description . . . . .	1174
8.1200.2Field Documentation . . . . .	1174
8.1200.2.1SlqsProfile3GPP . . . . .	1174
8.1200.2.2SlqsProfile3GPP2 . . . . .	1174
8.1201.1ds_RXBytesOKTlv Struct Reference . . . . .	1175
8.1201.1Detailed Description . . . . .	1175
8.1201.2Field Documentation . . . . .	1175
8.1201.2.1RxByteOKCnt . . . . .	1175
8.1201.2.2TlvPresent . . . . .	1175
8.1202.1ds_sourceOfChange Struct Reference . . . . .	1175
8.1202.1Detailed Description . . . . .	1175
8.1202.2Field Documentation . . . . .	1175
8.1202.2.1source . . . . .	1175
8.1203.1ds_transferStatInd Struct Reference . . . . .	1175
8.1203.1Detailed Description . . . . .	1176
8.1203.2Field Documentation . . . . .	1176
8.1203.2.1StatsMask . . . . .	1176
8.1203.2.2StatsPeriod . . . . .	1176
8.1204.1ds_TrStatInd Struct Reference . . . . .	1176
8.1204.1Detailed Description . . . . .	1176
8.1204.2Field Documentation . . . . .	1177
8.1204.2.1statsMask . . . . .	1177
8.1204.2.2statsPeriod . . . . .	1177
8.1205.1ds_TXBytesOKTlv Struct Reference . . . . .	1177
8.1205.1Detailed Description . . . . .	1177
8.1205.2Field Documentation . . . . .	1177
8.1205.2.1TlvPresent . . . . .	1177
8.1205.2.2TxByteOKCnt . . . . .	1177
8.1206.1ds_UMTSMInQoS Struct Reference . . . . .	1177
8.1206.1Detailed Description . . . . .	1177
8.1206.2Field Documentation . . . . .	1179
8.1206.2.1deliveryErrSDU . . . . .	1179
8.1206.2.2grntDownlinkBitrate . . . . .	1179

8.1206.2.3	grntUplinkBitrate	1179
8.1206.2.4	maxDownlinkBitrate	1179
8.1206.2.5	maxSDUSize	1179
8.1206.2.6	maxUplinkBitrate	1179
8.1206.2.7	qosDeliveryOrder	1179
8.1206.2.8	resBerRatio	1179
8.1206.2.9	sduErrorRatio	1179
8.1206.2.10	trafficClass	1179
8.1206.2.11	trafficPriority	1179
8.1206.2.12	transferDelay	1179
8.1207	WdsDhcpv4HwConfig Struct Reference	1179
8.1207.1	Detailed Description	1180
8.1207.2	Field Documentation	1180
8.1207.2.1	chaddr	1180
8.1207.2.2	chaddrLen	1180
8.1207.2.3	hwType	1180
8.1208	WdsDhcpv4Option Struct Reference	1180
8.1208.1	Detailed Description	1180
8.1208.2	Field Documentation	1180
8.1208.2.1	optCode	1181
8.1208.2.2	optVal	1181
8.1208.2.3	optValLen	1181
8.1209	WdsDhcpv4OptionList Struct Reference	1181
8.1209.1	Detailed Description	1181
8.1209.2	Field Documentation	1181
8.1209.2.1	numOpt	1181
8.1209.2.2	optList	1181
8.1210	WdsDhcpv4ProfileId Struct Reference	1181
8.1210.1	Detailed Description	1181
8.1210.2	Field Documentation	1182
8.1210.2.1	profileId	1182
8.1210.2.2	profileType	1182
<b>9</b>	<b>File Documentation</b>	<b>1183</b>
9.1	apdoxypages.c File Reference	1183
9.1.1	Detailed Description	1183
9.2	audio.h File Reference	1183
9.2.1	Function Documentation	1184
9.2.1.1	pack_audio_SLQSGetAudioPathConfig	1184
9.2.1.2	pack_audio_SLQSGetAudioProfile	1184

9.2.1.3	<a href="#">pack_audio_SLQSGetAudioVolTLBConfig</a>	1185
9.2.1.4	<a href="#">pack_audio_SLQSSetAudioPathConfig</a>	1185
9.2.1.5	<a href="#">pack_audio_SLQSSetAudioProfile</a>	1186
9.2.1.6	<a href="#">pack_audio_SLQSSetAudioVolTLBConfig</a>	1186
9.2.1.7	<a href="#">unpack_audio_SLQSGetAudioPathConfig</a>	1187
9.2.1.8	<a href="#">unpack_audio_SLQSGetAudioProfile</a>	1187
9.2.1.9	<a href="#">unpack_audio_SLQSGetAudioVolTLBConfig</a>	1187
9.2.1.10	<a href="#">unpack_audio_SLQSSetAudioPathConfig</a>	1188
9.2.1.11	<a href="#">unpack_audio_SLQSSetAudioProfile</a>	1188
9.2.1.12	<a href="#">unpack_audio_SLQSSetAudioVolTLBConfig</a>	1188
9.3	<a href="#">cat.h File Reference</a>	1189
9.3.1	<a href="#">Macro Definition Documentation</a>	1190
9.3.1.1	<a href="#">CAN_COMMON_EVENT_TLV_NUMBER</a>	1190
9.3.1.2	<a href="#">CAT_EVENT_DATA_MAX_LENGTH</a>	1190
9.3.2	<a href="#">Typedef Documentation</a>	1190
9.3.2.1	<a href="#">unpack_cat_CATSendEnvelopeCommand_t</a>	1190
9.3.2.2	<a href="#">unpack_cat_CATSendTerminalResponse_t</a>	1190
9.3.3	<a href="#">Function Documentation</a>	1190
9.3.3.1	<a href="#">pack_cat_CATSendEnvelopeCommand</a>	1190
9.3.3.2	<a href="#">pack_cat_CATSendTerminalResponse</a>	1190
9.3.3.3	<a href="#">pack_cat_SetCATEventCallback</a>	1191
9.3.3.4	<a href="#">unpack_cat_CATSendEnvelopeCommand</a>	1191
9.3.3.5	<a href="#">unpack_cat_CATSendTerminalResponse</a>	1191
9.3.3.6	<a href="#">unpack_cat_SetCATEventCallback</a>	1192
9.3.3.7	<a href="#">unpack_cat_SetCatEventCallback_ind</a>	1192
9.4	<a href="#">common.h File Reference</a>	1192
9.4.1	<a href="#">Macro Definition Documentation</a>	1194
9.4.1.1	<a href="#">DEFAULT_LOC_TIMEOUT_IN_SEC</a>	1194
9.4.1.2	<a href="#">MINREQBKLEN</a>	1194
9.4.1.3	<a href="#">MSGID_AND_LEN</a>	1194
9.4.1.4	<a href="#">MSGID_DONT_CARE</a>	1194
9.4.1.5	<a href="#">SDK_VALIDATE_INPUT_PACK_PARAM</a>	1194
9.4.1.6	<a href="#">SDK_VALIDATE_INPUT_PACK_PARAM_AND_FILL_XID</a>	1194
9.4.1.7	<a href="#">SDK_VALIDATE_INPUT_UNPACK_PARAM</a>	1195
9.4.1.8	<a href="#">SDU_HDR_LEN</a>	1195
9.4.1.9	<a href="#">SWI_INIT_UNPACK_RESULT_VALUE</a>	1195
9.4.1.10	<a href="#">UNUSEDPARAM</a>	1195
9.4.2	<a href="#">Enumeration Type Documentation</a>	1195
9.4.2.1	<a href="#">eLOG_LEVEL</a>	1195
9.4.2.2	<a href="#">eQMI_SVC</a>	1195

9.4.2.3	eTimeout	1196
9.4.2.4	msgtype	1196
9.4.3	Function Documentation	1196
9.4.3.1	fill_pack_ctx	1196
9.4.3.2	fill_sdu_hdr	1196
9.4.3.3	get_version	1196
9.4.3.4	helper_get_error_code	1197
9.4.3.5	helper_get_error_reason	1197
9.4.3.6	helper_get_req_str	1197
9.4.3.7	helper_get_resp_ctx	1197
9.4.3.8	helper_get_xid	1197
9.4.3.9	helper_isBootLoader_DebugEnabled	1197
9.4.3.10	helper_set_log_func	1198
9.4.3.11	helper_set_log_lvl	1198
9.4.3.12	liteqmi_GetVersion	1198
9.4.3.13	liteqmi_helper_decode7bitAsciiEncString	1198
9.4.3.14	liteqmi_log	1198
9.4.3.15	unpack_result_code_only	1198
9.4.4	Variable Documentation	1198
9.4.4.1	glog	1198
9.4.4.2	gloglvl	1198
9.5	dms.h File Reference	1198
9.5.1	Macro Definition Documentation	1206
9.5.1.1	ACT_CODE_MAX_SIZE	1206
9.5.1.2	CK_MAX_SIZE	1206
9.5.1.3	DMS_IMGDETAILS_LEN	1206
9.5.1.4	DMS_MAX_CUST_ID_LEN	1206
9.5.1.5	DMS_MAX_CUST_VALUE_LEN	1206
9.5.1.6	DMS_MAX_FWUPDATE_LOG_STR_SZ	1206
9.5.1.7	DMS_MAX_FWUPDATE_REF_STR_SZ	1206
9.5.1.8	DMS_MAX_RADIO_IFCS_SIZE	1206
9.5.1.9	DMS_MAX_SUBS_CFG_LIST_SIZE	1206
9.5.1.10	DMS_MAX_SUBS_LIST_SIZE	1206
9.5.1.11	DMS_MAX_SUPPORTED_LTE_BANDS	1206
9.5.1.12	DMS_PM_FACTORY	1207
9.5.1.13	DMS_PM_LOW	1207
9.5.1.14	DMS_PM_OFFLINE	1207
9.5.1.15	DMS_PM_ONLINE	1207
9.5.1.16	DMS_PM_PERSISTENT_LOW	1207
9.5.1.17	DMS_PM_RESET	1207



9.5.1.18	DMS_PM_SHUT_DOWN	1207
9.5.1.19	DMS_SET_REG_IND_DISABLE	1207
9.5.1.20	DMS_SET_REG_IND_ENABLE	1207
9.5.1.21	DMS_SET_REG_IND_NO_CHANGE	1207
9.5.1.22	DMS_SET_REPORT_DISABLE	1207
9.5.1.23	DMS_SET_REPORT_ENABLE	1207
9.5.1.24	DMS_SLQSFWINFO_APPVERSION_SZ	1207
9.5.1.25	DMS_SLQSFWINFO_BOOTVERSION_SZ	1207
9.5.1.26	DMS_SLQSFWINFO_CARRIER_SZ	1207
9.5.1.27	DMS_SLQSFWINFO_CUR_CARR_NAME	1207
9.5.1.28	DMS_SLQSFWINFO_CUR_CARR_REV	1207
9.5.1.29	DMS_SLQSFWINFO_MODELID_SZ	1207
9.5.1.30	DMS_SLQSFWINFO_PACKAGEID_SZ	1207
9.5.1.31	DMS_SLQSFWINFO_PRIVERSION_SZ	1207
9.5.1.32	DMS_SLQSFWINFO_SKU_SZ	1207
9.5.1.33	DMS_SWI_SET_IND_DISABLE	1207
9.5.1.34	DMS_SWI_SET_IND_ENABLE	1207
9.5.1.35	DMS_UINT8_MAX_STRING_SZ	1207
9.5.1.36	DMS_VALID_FSN_LEN	1207
9.5.1.37	ERI_DATA_MAX_SIZE	1207
9.5.1.38	MAX_BUILD_ID_LEN	1207
9.5.1.39	MEID_MAX_SIZE	1207
9.5.1.40	SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH	1208
9.5.1.41	SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH	1208
9.5.1.42	SPC_SIZE	1208
9.5.1.43	UNIQUE_ID_LEN	1208
9.5.2	Function Documentation	1208
9.5.2.1	pack_dms_ActivateAutomatic	1208
9.5.2.2	pack_dms_GetActivationState	1208
9.5.2.3	pack_dms_GetBandCapability	1208
9.5.2.4	pack_dms_GetCrashAction	1209
9.5.2.5	pack_dms_GetCustFeature	1209
9.5.2.6	pack_dms_GetCustFeaturesV2	1210
9.5.2.7	pack_dms_GetDeviceCap	1210
9.5.2.8	pack_dms_GetDeviceCapabilities	1210
9.5.2.9	pack_dms_GetDeviceCapabilitiesV2	1211
9.5.2.10	pack_dms_GetDeviceHardwareRev	1211
9.5.2.11	pack_dms_GetDeviceMfr	1211
9.5.2.12	pack_dms_GetDeviceSerialNumbers	1212
9.5.2.13	pack_dms_GetFirmwareInfo	1212

9.5.2.14	<a href="#">pack_dms_GetFirmwareRevision</a>	1212
9.5.2.15	<a href="#">pack_dms_GetFirmwareRevisions</a>	1213
9.5.2.16	<a href="#">pack_dms_GetFSN</a>	1213
9.5.2.17	<a href="#">pack_dms_GetHardwareRevision</a>	1213
9.5.2.18	<a href="#">pack_dms_GetIMSI</a>	1214
9.5.2.19	<a href="#">pack_dms_GetManufacturer</a>	1214
9.5.2.20	<a href="#">pack_dms_GetModelID</a>	1214
9.5.2.21	<a href="#">pack_dms_GetNetworkTime</a>	1215
9.5.2.22	<a href="#">pack_dms_GetNetworkTimeV2</a>	1215
9.5.2.23	<a href="#">pack_dms_GetOfflineReason</a>	1216
9.5.2.24	<a href="#">pack_dms_GetPower</a>	1216
9.5.2.25	<a href="#">pack_dms_GetPRLVersion</a>	1216
9.5.2.26	<a href="#">pack_dms_GetSerialNumbers</a>	1217
9.5.2.27	<a href="#">pack_dms_GetUSBComp</a>	1217
9.5.2.28	<a href="#">pack_dms_GetVoiceNumber</a>	1217
9.5.2.29	<a href="#">pack_dms_ResetToFactoryDefaults</a>	1218
9.5.2.30	<a href="#">pack_dms_SetActivationStatusCallback</a>	1218
9.5.2.31	<a href="#">pack_dms_SetCrashAction</a>	1218
9.5.2.32	<a href="#">pack_dms_SetCustFeature</a>	1219
9.5.2.33	<a href="#">pack_dms_SetCustFeaturesV2</a>	1219
9.5.2.34	<a href="#">pack_dms_SetEventReport</a>	1219
9.5.2.35	<a href="#">pack_dms_SetFirmwarePreference</a>	1220
9.5.2.36	<a href="#">pack_dms_SetIndicationRegister</a>	1220
9.5.2.37	<a href="#">pack_dms_SetPower</a>	1221
9.5.2.38	<a href="#">pack_dms_SetUSBComp</a>	1221
9.5.2.39	<a href="#">pack_dms_SLQSDmsSwiGetPCInfo</a>	1221
9.5.2.40	<a href="#">pack_dms_SLQSDmsSwiGetResetInfo</a>	1222
9.5.2.41	<a href="#">pack_dms_SLQSDmsSwiGetUimSelection</a>	1222
9.5.2.42	<a href="#">pack_dms_SLQSDmsSwiIndicationRegister</a>	1222
9.5.2.43	<a href="#">pack_dms_SLQSGetBandCapability</a>	1223
9.5.2.44	<a href="#">pack_dms_SLQSGetERIFile</a>	1223
9.5.2.45	<a href="#">pack_dms_SLQSGetPowerSaveModeConfig</a>	1224
9.5.2.46	<a href="#">pack_dms_SLQSSetPowerSaveModeConfig</a>	1224
9.5.2.47	<a href="#">pack_dms_SLQSSwiClearDyingGaspStatistics</a>	1224
9.5.2.48	<a href="#">pack_dms_SLQSSwiGetCrashInfo</a>	1225
9.5.2.49	<a href="#">pack_dms_SLQSSwiGetDyingGaspCfg</a>	1225
9.5.2.50	<a href="#">pack_dms_SLQSSwiGetDyingGaspStatistics</a>	1225
9.5.2.51	<a href="#">pack_dms_SLQSSwiGetFirmwareCurr</a>	1226
9.5.2.52	<a href="#">pack_dms_SLQSSwiGetFwUpdateStatus</a>	1226
9.5.2.53	<a href="#">pack_dms_SLQSSwiGetHostDevInfo</a>	1226

9.5.2.54	pack_dms_SLQSSwiGetOSInfo	1227
9.5.2.55	pack_dms_SLQSSwiGetSerialNoExt	1227
9.5.2.56	pack_dms_SLQSSwiSetDyingGaspCfg	1227
9.5.2.57	pack_dms_SLQSSwiSetHostDevInfo	1228
9.5.2.58	pack_dms_SLQSSwiSetOSInfo	1228
9.5.2.59	pack_dms_SLQSUIMGetState	1228
9.5.2.60	pack_dms_SwiSetEventReport	1229
9.5.2.61	pack_dms_SwiUimSelect	1229
9.5.2.62	pack_dms_UIMChangePIN	1230
9.5.2.63	pack_dms_UIMGetControlKeyStatus	1230
9.5.2.64	pack_dms_UIMGetICCID	1230
9.5.2.65	pack_dms_UIMGetPINStatus	1231
9.5.2.66	pack_dms_UIMSetControlKeyProtection	1231
9.5.2.67	pack_dms_UIMSetPINProtection	1231
9.5.2.68	pack_dms_UIMUnblockControlKey	1232
9.5.2.69	pack_dms_UIMUnblockPIN	1232
9.5.2.70	pack_dms_UIMVerifyPIN	1233
9.5.2.71	pack_dms_ValidateSPC	1233
9.5.2.72	unpack_dms_ActivateAutomatic	1233
9.5.2.73	unpack_dms_GetActivationState	1234
9.5.2.74	unpack_dms_GetBandCapability	1234
9.5.2.75	unpack_dms_GetCrashAction	1234
9.5.2.76	unpack_dms_GetCustFeature	1235
9.5.2.77	unpack_dms_GetCustFeaturesV2	1235
9.5.2.78	unpack_dms_GetDeviceCap	1235
9.5.2.79	unpack_dms_GetDeviceCapabilities	1236
9.5.2.80	unpack_dms_GetDeviceCapabilitiesV2	1236
9.5.2.81	unpack_dms_GetDeviceHardwareRev	1236
9.5.2.82	unpack_dms_GetDeviceMfr	1237
9.5.2.83	unpack_dms_GetDeviceSerialNumbers	1237
9.5.2.84	unpack_dms_GetFirmwareInfo	1237
9.5.2.85	unpack_dms_GetFirmwareRevision	1238
9.5.2.86	unpack_dms_GetFirmwareRevisions	1238
9.5.2.87	unpack_dms_GetFSN	1238
9.5.2.88	unpack_dms_GetHardwareRevision	1239
9.5.2.89	unpack_dms_GetIMSI	1239
9.5.2.90	unpack_dms_GetManufacturer	1239
9.5.2.91	unpack_dms_GetModelID	1240
9.5.2.92	unpack_dms_GetNetworkTime	1240
9.5.2.93	unpack_dms_GetNetworkTimeV2	1240

9.5.2.94	<a href="#">unpack_dms_GetOfflineReason</a>	1241
9.5.2.95	<a href="#">unpack_dms_GetPower</a>	1241
9.5.2.96	<a href="#">unpack_dms_GetPRLVersion</a>	1241
9.5.2.97	<a href="#">unpack_dms_GetSerialNumbers</a>	1242
9.5.2.98	<a href="#">unpack_dms_GetUSBComp</a>	1242
9.5.2.99	<a href="#">unpack_dms_GetVoiceNumber</a>	1243
9.5.2.100	<a href="#">unpack_dms_PSMCfgChange_ind</a>	1243
9.5.2.101	<a href="#">unpack_dms_ResetToFactoryDefaults</a>	1243
9.5.2.102	<a href="#">unpack_dms_SetActivationStatusCallback</a>	1244
9.5.2.103	<a href="#">unpack_dms_SetCrashAction</a>	1244
9.5.2.104	<a href="#">unpack_dms_SetCustFeature</a>	1244
9.5.2.105	<a href="#">unpack_dms_SetCustFeaturesV2</a>	1245
9.5.2.106	<a href="#">unpack_dms_SetEventReport</a>	1245
9.5.2.107	<a href="#">unpack_dms_SetEventReport_ind</a>	1245
9.5.2.108	<a href="#">unpack_dms_SetFirmwarePreference</a>	1246
9.5.2.109	<a href="#">unpack_dms_SetIndicationRegister</a>	1246
9.5.2.110	<a href="#">unpack_dms_SetPower</a>	1246
9.5.2.111	<a href="#">unpack_dms_SetUSBComp</a>	1247
9.5.2.112	<a href="#">unpack_dms_SLQSDmsSwiGetPCInfo</a>	1247
9.5.2.113	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo</a>	1247
9.5.2.114	<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_Ind</a>	1248
9.5.2.115	<a href="#">unpack_dms_SLQSDmsSwiGetUimSelection</a>	1248
9.5.2.116	<a href="#">unpack_dms_SLQSDmsSwiIndicationRegister</a>	1249
9.5.2.117	<a href="#">unpack_dms_SLQSGetBandCapability</a>	1249
9.5.2.118	<a href="#">unpack_dms_SLQSGetBandCapabilityExt</a>	1249
9.5.2.119	<a href="#">unpack_dms_SLQSGetERIFile</a>	1250
9.5.2.120	<a href="#">unpack_dms_SLQSGetPowerSaveModeConfig</a>	1250
9.5.2.121	<a href="#">unpack_dms_SLQSSetPowerSaveModeConfig</a>	1250
9.5.2.122	<a href="#">unpack_dms_SLQSSwiClearDyingGaspStatistics</a>	1251
9.5.2.123	<a href="#">unpack_dms_SLQSSwiGetCrashInfo</a>	1251
9.5.2.124	<a href="#">unpack_dms_SLQSSwiGetDyingGaspCfg</a>	1251
9.5.2.125	<a href="#">unpack_dms_SLQSSwiGetDyingGaspStatistics</a>	1252
9.5.2.126	<a href="#">unpack_dms_SLQSSwiGetFirmwareCurr</a>	1252
9.5.2.127	<a href="#">unpack_dms_SLQSSwiGetFwUpdateStatus</a>	1252
9.5.2.128	<a href="#">unpack_dms_SLQSSwiGetHostDevInfo</a>	1253
9.5.2.129	<a href="#">unpack_dms_SLQSSwiGetOSInfo</a>	1253
9.5.2.130	<a href="#">unpack_dms_SLQSSwiGetSerialNoExt</a>	1253
9.5.2.131	<a href="#">unpack_dms_SLQSSwiSetDyingGaspCfg</a>	1254
9.5.2.132	<a href="#">unpack_dms_SLQSSwiSetHostDevInfo</a>	1254
9.5.2.133	<a href="#">unpack_dms_SLQSSwiSetOSInfo</a>	1254

9.5.2.134	unpack_dms_SLQSUIGetState	1255
9.5.2.135	unpack_dms_SwiEventReportCallBack_ind	1255
9.5.2.136	unpack_dms_SwiSetEventReport	1255
9.5.2.137	unpack_dms_SwiUimSelect	1256
9.5.2.138	unpack_dms_UIMChangePIN	1256
9.5.2.139	unpack_dms_UIMGetControlKeyStatus	1256
9.5.2.140	unpack_dms_UIMGetICCID	1257
9.5.2.141	unpack_dms_UIMGetPINStatus	1257
9.5.2.142	unpack_dms_UIMSetControlKeyProtection	1257
9.5.2.143	unpack_dms_UIMSetPINProtection	1258
9.5.2.144	unpack_dms_UIMUnblockControlKey	1258
9.5.2.145	unpack_dms_UIMUnblockPIN	1259
9.5.2.146	unpack_dms_UIMVerifyPIN	1259
9.5.2.147	unpack_dms_ValidateSPC	1259
9.6	fms.h File Reference	1260
9.6.1	Macro Definition Documentation	1261
9.6.1.1	FMS_FW_PRI_BUILD_MATCH_LEN	1261
9.6.1.2	FMS_GOBI_LISTENTRIES_MAX	1261
9.6.1.3	FMS_GOBI_MBN_BUILD_ID_STR_LEN	1261
9.6.1.4	FMS_GOBI_MBN_IMG_ID_SIZE	1261
9.6.1.5	FMS_GOBI_MBN_IMG_ID_STR_LEN	1261
9.6.1.6	FMS_IMAGE_ID_BUILD_ID_LEN	1261
9.6.1.7	FMS_IMAGE_ID_IMG_ID_LEN	1261
9.6.1.8	FMS_IMAGE_ID_MAX_ENTRIES	1261
9.6.1.9	FMS_IMAGE_ID_PRI_IMGTYPE	1261
9.6.1.10	FMS_MAX_IMAGE_ID_ELEMENT	1261
9.6.1.11	FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE	1261
9.6.1.12	LITE_TOTAL_IMAGE_ID_STRING_SIZE	1261
9.6.2	Function Documentation	1261
9.6.2.1	GetValidFwPriCombinations	1261
9.6.2.2	pack_fms_GetImagesPreference	1262
9.6.2.3	pack_fms_GetStoredImages	1262
9.6.2.4	pack_fms_SetImagesPreference	1262
9.6.2.5	unpack_fms_GetImagesPreference	1262
9.6.2.6	unpack_fms_GetStoredImages	1262
9.6.2.7	unpack_fms_SetImagesPreference	1263
9.7	ims.h File Reference	1263
9.7.1	Macro Definition Documentation	1265
9.7.1.1	MAX_NAME_LEN	1265
9.7.2	Typedef Documentation	1265

9.7.2.1	<a href="#">unpack_ims_SLQSImsConfigIndicationRegister_t</a>	1265
9.7.3	<a href="#">Function Documentation</a>	1265
9.7.3.1	<a href="#">pack_ims_SLQSGetIMSSMSConfig</a>	1265
9.7.3.2	<a href="#">pack_ims_SLQSGetIMSUserConfig</a>	1265
9.7.3.3	<a href="#">pack_ims_SLQSGetIMSVoIPConfig</a>	1266
9.7.3.4	<a href="#">pack_ims_SLQSGetRegMgrConfig</a>	1266
9.7.3.5	<a href="#">pack_ims_SLQSGetSIPConfig</a>	1267
9.7.3.6	<a href="#">pack_ims_SLQSImsConfigIndicationRegister</a>	1267
9.7.3.7	<a href="#">pack_ims_SLQSSetIMSSMSConfig</a>	1268
9.7.3.8	<a href="#">pack_ims_SLQSSetIMSUserConfig</a>	1268
9.7.3.9	<a href="#">pack_ims_SLQSSetIMSVoIPConfig</a>	1269
9.7.3.10	<a href="#">pack_ims_SLQSSetRegMgrConfig</a>	1269
9.7.3.11	<a href="#">pack_ims_SLQSSetSIPConfig</a>	1270
9.7.3.12	<a href="#">unpack_ims_SLQSGetIMSSMSConfig</a>	1270
9.7.3.13	<a href="#">unpack_ims_SLQSGetIMSUserConfig</a>	1270
9.7.3.14	<a href="#">unpack_ims_SLQSGetIMSVoIPConfig</a>	1271
9.7.3.15	<a href="#">unpack_ims_SLQSGetRegMgrConfig</a>	1271
9.7.3.16	<a href="#">unpack_ims_SLQSGetSIPConfig</a>	1271
9.7.3.17	<a href="#">unpack_ims_SLQSImsConfigIndicationRegister</a>	1272
9.7.3.18	<a href="#">unpack_ims_SLQSRegMgrCfgCallBack_ind</a>	1272
9.7.3.19	<a href="#">unpack_ims_SLQSSetIMSSMSConfig</a>	1272
9.7.3.20	<a href="#">unpack_ims_SLQSSetIMSUserConfig</a>	1273
9.7.3.21	<a href="#">unpack_ims_SLQSSetIMSVoIPConfig</a>	1273
9.7.3.22	<a href="#">unpack_ims_SLQSSetRegMgrConfig</a>	1273
9.7.3.23	<a href="#">unpack_ims_SLQSSetSIPConfig</a>	1274
9.7.3.24	<a href="#">unpack_ims_SLQSSIPCfgCallBack_ind</a>	1274
9.7.3.25	<a href="#">unpack_ims_SLQSSMSCfgCallBack_ind</a>	1274
9.7.3.26	<a href="#">unpack_ims_SLQSUserCfgCallBack_ind</a>	1275
9.7.3.27	<a href="#">unpack_ims_SLQSVoIPCfgCallBack_ind</a>	1275
9.8	<a href="#">imsa.h File Reference</a>	1275
9.8.1	<a href="#">Macro Definition Documentation</a>	1277
9.8.1.1	<a href="#">MAX_ERROR_CODE_LEN</a>	1277
9.8.2	<a href="#">Typedef Documentation</a>	1277
9.8.2.1	<a href="#">unpack_imsa_SLQSRegisterIMSAIndication_t</a>	1277
9.8.3	<a href="#">Function Documentation</a>	1277
9.8.3.1	<a href="#">pack_imsa_SLQSGetIMSARegStatus</a>	1277
9.8.3.2	<a href="#">pack_imsa_SLQSGetIMSAServiceStatus</a>	1277
9.8.3.3	<a href="#">pack_imsa_SLQSRegisterIMSAIndication</a>	1278
9.8.3.4	<a href="#">unpack_imsa_SLQSGetIMSARegStatus</a>	1278
9.8.3.5	<a href="#">unpack_imsa_SLQSGetIMSAServiceStatus</a>	1279

9.8.3.6	<a href="#">unpack_imsa_SLQSImsaPdpStatusCallBack_ind</a>	1279
9.8.3.7	<a href="#">unpack_imsa_SLQSImsaRatStatusCallBack_ind</a>	1280
9.8.3.8	<a href="#">unpack_imsa_SLQSImsaRegStatusCallBack_ind</a>	1280
9.8.3.9	<a href="#">unpack_imsa_SLQSImsaSvcStatusCallBack_ind</a>	1280
9.8.3.10	<a href="#">unpack_imsa_SLQSRegisterIMSAIndication</a>	1281
9.9	<a href="#">lite-fw.h File Reference</a>	1281
9.9.1	<a href="#">Detailed Description</a>	1284
9.9.2	<a href="#">Macro Definition Documentation</a>	1284
9.9.2.1	<a href="#">FIRMWARE_BCHDATESIZE</a>	1284
9.9.2.2	<a href="#">FIRMWARE_BCHVERSTRSIZE</a>	1284
9.9.2.3	<a href="#">FIRMWARE_IMAGE_SIZE_MAX</a>	1284
9.9.2.4	<a href="#">FIRMWARE_INFO_STRING_SIZE</a>	1284
9.9.2.5	<a href="#">IMG_MASK_CLEAR</a>	1284
9.9.2.6	<a href="#">IMG_MASK_GENERIC</a>	1284
9.9.2.7	<a href="#">IMG_MASK_MDM</a>	1284
9.9.2.8	<a href="#">IMG_MASK_PRI</a>	1284
9.9.2.9	<a href="#">libSDP_BuildImagesPreferenceRequest</a>	1284
9.9.2.10	<a href="#">libSDP_CalculateImageMask</a>	1284
9.9.2.11	<a href="#">LIBSDP_CARRIER_PACKAGE_SKU</a>	1284
9.9.2.12	<a href="#">libSDP_CheckValidFirmwareInfo</a>	1284
9.9.2.13	<a href="#">libSDP_DownloadFW</a>	1284
9.9.2.14	<a href="#">libSDP_ExtractFirmwareParametersByPath</a>	1284
9.9.2.15	<a href="#">libSDP_getFileType</a>	1284
9.9.2.16	<a href="#">libSDP_GetModelFamily</a>	1284
9.9.2.17	<a href="#">libSDP_GetVersion</a>	1284
9.9.2.18	<a href="#">libsdp_set_log_func</a>	1284
9.9.2.19	<a href="#">libsdp_SetReadBlockSize</a>	1284
9.9.2.20	<a href="#">LIBSDP_SKU_STRING_LENGTH</a>	1284
9.9.2.21	<a href="#">LITEFW_CARRIER_PACKAGE_SKU</a>	1284
9.9.2.22	<a href="#">LITEFW_SKU_STRING_LENGTH</a>	1284
9.9.2.23	<a href="#">MAX_IMAGE_PRODUCT_LENGTH</a>	1284
9.9.2.24	<a href="#">NULL_TERMINATOR_CHAR_SIZE</a>	1285
9.9.3	<a href="#">Typedef Documentation</a>	1285
9.9.3.1	<a href="#">libSDP_FirmwareInfo</a>	1285
9.9.3.2	<a href="#">libsdplogger</a>	1285
9.9.3.3	<a href="#">litefw_FirmwareFileInfo</a>	1285
9.9.3.4	<a href="#">litefw_FirmwareInfo</a>	1286
9.9.3.5	<a href="#">litefw_FirmwarePartNo</a>	1286
9.9.3.6	<a href="#">litefwlogger</a>	1286
9.9.4	<a href="#">Enumeration Type Documentation</a>	1286

9.9.4.1	<a href="#">litefw_fileimgtype</a>	1287
9.9.4.2	<a href="#">litefw_Fw_Type</a>	1287
9.9.4.3	<a href="#">litefw_fwdwl_error_codes</a>	1287
9.9.4.4	<a href="#">litefw_imagetype</a>	1288
9.9.4.5	<a href="#">litefw_Models</a>	1288
9.9.4.6	<a href="#">litefw_QDL_FLOW_CONTROLS</a>	1288
9.9.4.7	<a href="#">litefw_QDL_MODEs</a>	1288
9.9.5	<a href="#">Function Documentation</a>	1289
9.9.5.1	<a href="#">litefw_BuildImagesPreferenceRequest</a>	1289
9.9.5.2	<a href="#">litefw_CalculateImageMask</a>	1289
9.9.5.3	<a href="#">litefw_CheckValidFirmwareInfo</a>	1289
9.9.5.4	<a href="#">litefw_DownloadFW</a>	1290
9.9.5.5	<a href="#">litefw_ExtractFirmwareParametersByPath</a>	1290
9.9.5.6	<a href="#">litefw_ExtractFirmwarePartNoByPath</a>	1291
9.9.5.7	<a href="#">litefw_getFileType</a>	1291
9.9.5.8	<a href="#">litefw_GetModelFamily</a>	1291
9.9.5.9	<a href="#">litefw_GetQTLDownloadMode</a>	1292
9.9.5.10	<a href="#">litefw_GetQTLHWFlowControl</a>	1292
9.9.5.11	<a href="#">litefw_GetVersion</a>	1292
9.9.5.12	<a href="#">litefw_logsenable</a>	1292
9.9.5.13	<a href="#">litefw_set_log_func</a>	1292
9.9.5.14	<a href="#">litefw_SetQTLDownloadMode</a>	1293
9.9.5.15	<a href="#">litefw_SetQTLHWFlowControl</a>	1293
9.9.5.16	<a href="#">litefw_SetReadBlockSize</a>	1293
9.9.5.17	<a href="#">litefw_SLQSGetFirmwareFileInfo</a>	1293
9.9.5.18	<a href="#">litefw_switch_9x07_to_downloadmode</a>	1294
9.9.5.19	<a href="#">litefw_switch_to_BootHoldMode</a>	1294
9.10	<a href="#">loc.h File Reference</a>	1294
9.10.1	<a href="#">Macro Definition Documentation</a>	1298
9.10.1.1	<a href="#">LOC_UINT8_MAX_STRING_SZ</a>	1298
9.10.1.2	<a href="#">LOCEVENTMASKBATCHFULLNOTIFICATION</a>	1298
9.10.1.3	<a href="#">LOCEVENTMASKENGINESTATE</a>	1298
9.10.1.4	<a href="#">LOCEVENTMASKFIXSESSIONSTATE</a>	1299
9.10.1.5	<a href="#">LOCEVENTMASKGEOFENCEBATCHBREACHNOTIFICATION</a>	1299
9.10.1.6	<a href="#">LOCEVENTMASKGEOFENCEBREACHNOTIFICATION</a>	1299
9.10.1.7	<a href="#">LOCEVENTMASKGEOFENCEGENALERT</a>	1299
9.10.1.8	<a href="#">LOCEVENTMASKGNSSMEASUREMENTREPORT</a>	1299
9.10.1.9	<a href="#">LOCEVENTMASKGNSSSVINFO</a>	1299
9.10.1.10	<a href="#">LOCEVENTMASKINJECTPOSITIONREQ</a>	1299
9.10.1.11	<a href="#">LOCEVENTMASKINJECTPREDICTEDORBITSREQ</a>	1299



9.10.1.12	LOCEVENTMASKINJECTTIMERREQ	1299
9.10.1.13	LOCEVENTMASKINJECTWIFIAPDATAREQ	1299
9.10.1.14	LOCEVENTMASKINVALIDVALUE	1299
9.10.1.15	LOCEVENTMASKLIVEBATCHEDPOSITIONREPORT	1300
9.10.1.16	LOCEVENTMASKLOCATIONSERVERCONNECTIONREQ	1300
9.10.1.17	LOCEVENTMASKMOTIONDATACONTROL	1300
9.10.1.18	LOCEVENTMASKNIGEOFENCENOTIFICATION	1300
9.10.1.19	LOCEVENTMASKNINOTIFYVERIFYREQ	1300
9.10.1.20	LOCEVENTMASKNMEA	1300
9.10.1.21	LOCEVENTMASKPEDOMETERCONTROL	1300
9.10.1.22	LOCEVENTMASKPOSITIONREPORT	1300
9.10.1.23	LOCEVENTMASKSENSORSTREAMINGREADYSTATUS	1300
9.10.1.24	LOCEVENTMASKSETSPISTREAMINGREPORT	1300
9.10.1.25	LOCEVENTMASKTIMESYNCREQ	1301
9.10.1.26	LOCEVENTMASKVEHICLEDATAREADYSTATUS	1301
9.10.1.27	LOCEVENTMASKWIFIREQ	1301
9.10.1.28	MAX_LOC_NMEA_STR_LEN	1301
9.10.1.29	MAX_SENSOR_DATA_LEN	1301
9.10.1.30	MAX_TEMP_DATA_LEN	1301
9.10.2	Typedef Documentation	1301
9.10.2.1	unpack_loc_DeleteAssistData_t	1301
9.10.2.2	unpack_loc_GetFixCriteria_t	1301
9.10.2.3	unpack_loc_SLQSLOCGetServer_t	1301
9.10.2.4	unpack_loc_SLQSLOCInjectPosition_t	1301
9.10.2.5	unpack_loc_SLQSLOCInjectSensorData_t	1301
9.10.2.6	unpack_loc_SLQSLOCInjectUTCTime_t	1301
9.10.2.7	unpack_loc_SLQSLOCSetCradleMountConfig_t	1301
9.10.3	Enumeration Type Documentation	1301
9.10.3.1	anonymous enum	1301
9.10.4	Function Documentation	1301
9.10.4.1	pack_loc_DeleteAssistData	1301
9.10.4.2	pack_loc_EventRegister	1302
9.10.4.3	pack_loc_GetFixCriteria	1302
9.10.4.4	pack_loc_SetExtPowerState	1303
9.10.4.5	pack_loc_SetOperationMode	1303
9.10.4.6	pack_loc_SLQSLOCGetBestAvailPos	1303
9.10.4.7	pack_loc_SLQSLOCGetOpMode	1304
9.10.4.8	pack_loc_SLQSLOCGetServer	1304
9.10.4.9	pack_loc_SLQSLOCInjectPosition	1305
9.10.4.10	pack_loc_SLQSLOCInjectSensorData	1305

9.10.4.11 pack_loc_SLQSLOCInjectUTCTime . . . . .	1305
9.10.4.12 pack_loc_SLQSLOCSetCradleMountConfig . . . . .	1306
9.10.4.13 pack_loc_SLQSLOCSetServer . . . . .	1306
9.10.4.14 pack_loc_Start . . . . .	1307
9.10.4.15 pack_loc_Stop . . . . .	1307
9.10.4.16 unpack_loc_BestAvailPos_Ind . . . . .	1307
9.10.4.17 unpack_loc_CradleMountCallback_Ind . . . . .	1308
9.10.4.18 unpack_loc_DeleteAssistData . . . . .	1308
9.10.4.19 unpack_loc_DeleteAssistData_Ind . . . . .	1308
9.10.4.20 unpack_loc_EngineState_Ind . . . . .	1309
9.10.4.21 unpack_loc_EventNMEA_Ind . . . . .	1309
9.10.4.22 unpack_loc_EventRegister . . . . .	1309
9.10.4.23 unpack_loc_EventTimeSyncCallback_Ind . . . . .	1310
9.10.4.24 unpack_loc_FixCriteria_Ind . . . . .	1310
9.10.4.25 unpack_loc_GetFixCriteria . . . . .	1310
9.10.4.26 unpack_loc_GetOpMode_Ind . . . . .	1311
9.10.4.27 unpack_loc_GetServer_Ind . . . . .	1311
9.10.4.28 unpack_loc_GnssSvInfo_Ind . . . . .	1311
9.10.4.29 unpack_loc_InjectPositionCallback_Ind . . . . .	1312
9.10.4.30 unpack_loc_InjectSensorDataCallback_Ind . . . . .	1312
9.10.4.31 unpack_loc_InjectTimeSyncDataCallback_Ind . . . . .	1312
9.10.4.32 unpack_loc_InjectUTCTimeCallback_Ind . . . . .	1313
9.10.4.33 unpack_loc_PositionRpt_Ind . . . . .	1313
9.10.4.34 unpack_loc_SensorStreamingCallback_Ind . . . . .	1313
9.10.4.35 unpack_loc_SetExtPowerConfig_Ind . . . . .	1314
9.10.4.36 unpack_loc_SetExtPowerState . . . . .	1314
9.10.4.37 unpack_loc_SetOperationMode . . . . .	1314
9.10.4.38 unpack_loc_SetOperationMode_Ind . . . . .	1315
9.10.4.39 unpack_loc_SetServer_Ind . . . . .	1315
9.10.4.40 unpack_loc_SLQSLOCGetBestAvailPos . . . . .	1315
9.10.4.41 unpack_loc_SLQSLOCGetOpMode . . . . .	1316
9.10.4.42 unpack_loc_SLQSLOCGetServer . . . . .	1316
9.10.4.43 unpack_loc_SLQSLOCInjectPosition . . . . .	1316
9.10.4.44 unpack_loc_SLQSLOCInjectSensorData . . . . .	1317
9.10.4.45 unpack_loc_SLQSLOCInjectUTCTime . . . . .	1317
9.10.4.46 unpack_loc_SLQSLOCSetCradleMountConfig . . . . .	1317
9.10.4.47 unpack_loc_SLQSLOCSetServer . . . . .	1318
9.10.4.48 unpack_loc_Start . . . . .	1318
9.10.4.49 unpack_loc_Stop . . . . .	1318
9.11 nas.h File Reference . . . . .	1319

9.11.1	Macro Definition Documentation	1331
9.11.1.1	LITE_MAX_NAS_3GPP2_MDN_LEN	1331
9.11.1.2	LITE_MAX_SLQS_NAS_NW_NAME_SRC_LENGTH	1331
9.11.1.3	LITE_NW_SCAN_LTE_OP_MODE_MAX_LENGTH	1331
9.11.1.4	NAS_IMSI_M_S1_LENGTH	1331
9.11.1.5	NAS_IMSI_M_S2_LENGTH	1331
9.11.1.6	NAS_MAX_DESCRIPTION_LENGTH	1331
9.11.1.7	NAS_MAX_NUM_NETWORKS	1331
9.11.1.8	NAS_MAX_PHY_CA_AGG_SCELL_ARRRY_SIZE	1331
9.11.1.9	NAS_MAX_SCC_RX_INFO_INSTANCES	1332
9.11.1.10	NAS_MAX_SLQS_NAS_PCI_INFO_LENGTH	1332
9.11.1.11	NAS_MAX_SLQS_NAS_PCI_INFO_PLMN_LENGTH	1332
9.11.1.12	NAS_MCC_MNC_INST_LENGTH	1332
9.11.1.13	NAS_NAM_NAME_LENGTH	1332
9.11.1.14	NAS_OTA_MESSAGE_MAX_BUF_SIZE	1332
9.11.1.15	NAS_PLMN_LENGTH	1332
9.11.1.16	NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST	1332
9.11.2	Typedef Documentation	1332
9.11.2.1	nas_LTEBandPrefExtTlv	1332
9.11.2.2	nas_NR5gBandPrefTlv	1332
9.11.2.3	unpack_nas_InitiateDomainAttach_t	1332
9.11.2.4	unpack_nas_SetACCOLC_t	1332
9.11.2.5	unpack_nas_SetCDMANetworkParameters_t	1332
9.11.2.6	unpack_nas_SetLURRejectCallback_t	1332
9.11.2.7	unpack_nas_SetRFInfoCallback_t	1332
9.11.2.8	unpack_nas_SLQSConfigSigInfo_t	1332
9.11.2.9	unpack_nas_SLQSInitiateNetworkRegistration_t	1332
9.11.2.10	unpack_nas_SLQSNasConfigSigInfo2_t	1332
9.11.2.11	unpack_nas_SLQSNasIndicationRegisterExt_t	1332
9.11.2.12	unpack_nas_SLQSNasIndicationRegisterV2_t	1332
9.11.2.13	unpack_nas_SLQSNASSeteDRXParams_t	1332
9.11.2.14	unpack_nas_SLQSNasSwiIndicationRegister_t	1332
9.11.2.15	unpack_nas_SLQSNASSwiSetChannelLock_t	1333
9.11.2.16	unpack_nas_SLQSSetBandPreference_t	1333
9.11.2.17	unpack_nas_SLQSSetSignalStrengthsCallback_t	1333
9.11.2.18	unpack_nas_SLQSSetSysSelectionPref_t	1333
9.11.2.19	unpack_nas_SLQSSetSysSelectionPrefExt_t	1333
9.11.2.20	unpack_nas_SLQSSwiPSDetach_t	1333
9.11.3	Enumeration Type Documentation	1333
9.11.3.1	LITE_TYPE_OF_SERVICE_DOMAIN	1333

9.11.3.2	LITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND	1333
9.11.3.3	LITEQMI_NAS_LTE_CPHY_CA_BW_NRB	1334
9.11.3.4	LITEQMI_NAS_LTE_CPHY_SCELL_STATE	1334
9.11.3.5	NAS_LTE_CPHY_CA_BW_NRB_LITE	1335
9.11.3.6	NAS_LTE_CPHY_SCELL_STATE_LITE	1335
9.11.4	Function Documentation	1335
9.11.4.1	pack_nas_GetACCOLC	1335
9.11.4.2	pack_nas_GetANAAAAAuthenticationStatus	1335
9.11.4.3	pack_nas_GetCDMANetworkParameters	1336
9.11.4.4	pack_nas_GetHomeNetwork	1336
9.11.4.5	pack_nas_GetHomeNetwork3GPP2	1336
9.11.4.6	pack_nas_GetNetworkPreference	1337
9.11.4.7	pack_nas_GetRFInfo	1337
9.11.4.8	pack_nas_GetServingNetwork	1337
9.11.4.9	pack_nas_GetServingNetworkCapabilities	1337
9.11.4.10	pack_nas_GetSignalStrengths	1338
9.11.4.11	pack_nas_InitiateDomainAttach	1338
9.11.4.12	pack_nas_PerformNetworkScan	1339
9.11.4.13	pack_nas_PerformNetworkScanPCI	1339
9.11.4.14	pack_nas_SetACCOLC	1339
9.11.4.15	pack_nas_SetCDMANetworkParameters	1340
9.11.4.16	pack_nas_SetLURejectCallback	1340
9.11.4.17	pack_nas_SetNetworkPreference	1340
9.11.4.18	pack_nas_SetRFInfoCallback	1341
9.11.4.19	pack_nas_SLQSConfigSigInfo	1341
9.11.4.20	pack_nas_SLQSGetErrorRate	1341
9.11.4.21	pack_nas_SLQSGetHomeNetwork	1342
9.11.4.22	pack_nas_SLqsGetLTECphyCAInfo	1342
9.11.4.23	pack_nas_SLQSGetNetworkTime	1342
9.11.4.24	pack_nas_SLQSGetOperatorNameData	1343
9.11.4.25	pack_nas_SLQSGetPLMNName	1343
9.11.4.26	pack_nas_SLQSGetServingSystem	1343
9.11.4.27	pack_nas_SLQSGetServingSystemV2	1344
9.11.4.28	pack_nas_SLQSGetSignalStrength	1344
9.11.4.29	pack_nas_SLQSGetSysInfo	1344
9.11.4.30	pack_nas_SLQSGetSysInfoV2	1345
9.11.4.31	pack_nas_SLQSGetSysSelectionPref	1345
9.11.4.32	pack_nas_SLQSGetSysSelectionPrefExt	1345
9.11.4.33	pack_nas_SLQSGetSysSelectionPrefExtV2	1346
9.11.4.34	pack_nas_SLQSInitiateNetworkRegistration	1346

9.11.4.35 pack_nas_SLQSNasConfigSigInfo2 . . . . .	1346
9.11.4.36 pack_nas_SLQSNasGet3GPP2Subscription . . . . .	1347
9.11.4.37 pack_nas_SLQSNasGetCellLocationInfo . . . . .	1347
9.11.4.38 pack_nas_SLQSNasGetCellLocationInfoV2 . . . . .	1347
9.11.4.39 pack_nas_SLQSNASGeteDRXParams . . . . .	1348
9.11.4.40 pack_nas_SLQSNASGeteDRXParamsExt . . . . .	1348
9.11.4.41 pack_nas_SLQSNASGetForbiddenNetworks . . . . .	1348
9.11.4.42 pack_nas_SLQSNasGetHDRColorCode . . . . .	1349
9.11.4.43 pack_nas_SLQSNasGetRFInfo . . . . .	1349
9.11.4.44 pack_nas_SLQSNasGetSigInfo . . . . .	1349
9.11.4.45 pack_nas_SLQSNasGetTxRxInfo . . . . .	1350
9.11.4.46 pack_nas_SLQSNasIndicationRegisterExt . . . . .	1350
9.11.4.47 pack_nas_SLQSNasIndicationRegisterV2 . . . . .	1351
9.11.4.48 pack_nas_SLQSNASSeteDRXParams . . . . .	1351
9.11.4.49 pack_nas_SLQSNASSwiGetChannelLock . . . . .	1351
9.11.4.50 pack_nas_SLQSNasSwiIndicationRegister . . . . .	1352
9.11.4.51 pack_nas_SLQSNasSwiModemStatus . . . . .	1352
9.11.4.52 pack_nas_SLQSNASSwiSetChannelLock . . . . .	1352
9.11.4.53 pack_nas_SLQSPerformNetworkScanV2 . . . . .	1353
9.11.4.54 pack_nas_SLQSSetBandPreference . . . . .	1353
9.11.4.55 pack_nas_SLQSSetSignalStrengthsCallback . . . . .	1354
9.11.4.56 pack_nas_SLQSSetSysSelectionPref . . . . .	1354
9.11.4.57 pack_nas_SLQSSetSysSelectionPrefExt . . . . .	1354
9.11.4.58 pack_nas_SLQSSwiGetHDRPersonality . . . . .	1355
9.11.4.59 pack_nas_SLQSSwiGetHDRProtSubtype . . . . .	1355
9.11.4.60 pack_nas_SLQSSwiGetHRPDStats . . . . .	1355
9.11.4.61 pack_nas_SLQSSwiGetLteCQI . . . . .	1356
9.11.4.62 pack_nas_SLQSSwiGetLteSccRxInfo . . . . .	1356
9.11.4.63 pack_nas_SLQSSwiNetworkDebug . . . . .	1356
9.11.4.64 pack_nas_SLQSSwiPSDetach . . . . .	1357
9.11.4.65 unpack_nas_GetACCOLC . . . . .	1357
9.11.4.66 unpack_nas_GetANAAAAuthenticationStatus . . . . .	1357
9.11.4.67 unpack_nas_GetCDMANetworkParameters . . . . .	1358
9.11.4.68 unpack_nas_GetHomeNetwork . . . . .	1358
9.11.4.69 unpack_nas_GetHomeNetwork3GPP2 . . . . .	1358
9.11.4.70 unpack_nas_GetNetworkPreference . . . . .	1359
9.11.4.71 unpack_nas_GetRFInfo . . . . .	1359
9.11.4.72 unpack_nas_GetServingNetwork . . . . .	1359
9.11.4.73 unpack_nas_GetServingNetworkCapabilities . . . . .	1360
9.11.4.74 unpack_nas_GetSignalStrengths . . . . .	1360

9.11.4.75 unpack_nas_InitiateDomainAttach . . . . .	1361
9.11.4.76 unpack_nas_PerformNetworkScan . . . . .	1361
9.11.4.77 unpack_nas_SetACCOLC . . . . .	1361
9.11.4.78 unpack_nas_SetCDMANetworkParameters . . . . .	1362
9.11.4.79 unpack_nas_SetDataCapabilitiesCallback_ind . . . . .	1362
9.11.4.80 unpack_nas_SetEventReportInd . . . . .	1362
9.11.4.81 unpack_nas_SetLURejectCallback . . . . .	1363
9.11.4.82 unpack_nas_SetNasLTECphyCalndCallback_ind . . . . .	1363
9.11.4.83 unpack_nas_SetNetworkPreference . . . . .	1363
9.11.4.84 unpack_nas_SetRFInfoCallback . . . . .	1364
9.11.4.85 unpack_nas_SetRoamingIndicatorCallback_ind . . . . .	1364
9.11.4.86 unpack_nas_SetServingSystemCallback_ind . . . . .	1364
9.11.4.87 unpack_nas_SLQSConfigSigInfo . . . . .	1365
9.11.4.88 unpack_nas_SLQSGetErrorRate . . . . .	1365
9.11.4.89 unpack_nas_SLQSGetHomeNetwork . . . . .	1365
9.11.4.90 unpack_nas_SLQSGetLTECphyCAInfo . . . . .	1366
9.11.4.91 unpack_nas_SLQSGetNetworkTime . . . . .	1366
9.11.4.92 unpack_nas_SLQSGetOperatorNameData . . . . .	1366
9.11.4.93 unpack_nas_SLQSGetPLMNName . . . . .	1367
9.11.4.94 unpack_nas_SLQSGetServingSystem . . . . .	1367
9.11.4.95 unpack_nas_SLQSGetServingSystemV2 . . . . .	1367
9.11.4.96 unpack_nas_SLQSGetSignalStrength . . . . .	1368
9.11.4.97 unpack_nas_SLQSGetSysInfo . . . . .	1368
9.11.4.98 unpack_nas_SLQSGetSysInfoV2 . . . . .	1369
9.11.4.99 unpack_nas_SLQSGetSysSelectionPref . . . . .	1369
9.11.4.100unpack_nas_SLQSGetSysSelectionPrefExt . . . . .	1369
9.11.4.101unpack_nas_SLQSGetSysSelectionPrefExtV2 . . . . .	1370
9.11.4.102unpack_nas_SLQSInitiateNetworkRegistration . . . . .	1370
9.11.4.103unpack_nas_SLQSNasConfigSigInfo2 . . . . .	1370
9.11.4.104unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind . . . . .	1371
9.11.4.105unpack_nas_SLQSNasGet3GPP2Subscription . . . . .	1371
9.11.4.106unpack_nas_SLQSNasGetCellLocationInfo . . . . .	1372
9.11.4.107unpack_nas_SLQSNasGetCellLocationInfoV2 . . . . .	1372
9.11.4.108unpack_nas_SLQSNASGeteDRXParams . . . . .	1372
9.11.4.109unpack_nas_SLQSNASGeteDRXParamsExt . . . . .	1373
9.11.4.110unpack_nas_SLQSNASGetForbiddenNetworks . . . . .	1373
9.11.4.111unpack_nas_SLQSNasGetHDRColorCode . . . . .	1373
9.11.4.112unpack_nas_SLQSNasGetRFInfo . . . . .	1374
9.11.4.113unpack_nas_SLQSNasGetSigInfo . . . . .	1374
9.11.4.114unpack_nas_SLQSNasGetTxRxInfo . . . . .	1374

9.11.4.115unpack_nas_SLQSNasIndicationRegisterExt . . . . .	1375
9.11.4.116unpack_nas_SLQSNasIndicationRegisterV2 . . . . .	1375
9.11.4.117unpack_nas_SLQSNasNetworkRejectCallback_Ind . . . . .	1376
9.11.4.118unpack_nas_SLQSNasNetworkTimeCallBack_ind . . . . .	1376
9.11.4.119unpack_nas_SLQSNasRFBandInfoCallback_Ind . . . . .	1376
9.11.4.120unpack_nas_SLQSNASSeteDRXParams . . . . .	1377
9.11.4.121unpack_nas_SLQSNasSigInfoCallback_ind . . . . .	1377
9.11.4.122unpack_nas_SLQSNASSwiGetChannelLock . . . . .	1377
9.11.4.123unpack_nas_SLQSNasSwiIndicationRegister . . . . .	1378
9.11.4.124unpack_nas_SLQSNasSwiModemStatus . . . . .	1378
9.11.4.125unpack_nas_SLQSNasSwiOTAMessageCallback_ind . . . . .	1378
9.11.4.126unpack_nas_SLQSNASSwiSetChannelLock . . . . .	1379
9.11.4.127unpack_nas_SLQSNasSysInfoCallback_ind . . . . .	1379
9.11.4.128unpack_nas_SLQSNasTimerCallback_ind . . . . .	1379
9.11.4.129unpack_nas_SLQSPerformNetworkScanV2 . . . . .	1380
9.11.4.130unpack_nas_SLQSSetBandPreference . . . . .	1380
9.11.4.131unpack_nas_SLQSSetSignalStrengthsCallback . . . . .	1380
9.11.4.132unpack_nas_SLQSSetSysSelectionPref . . . . .	1381
9.11.4.133unpack_nas_SLQSSetSysSelectionPrefCallBack_ind . . . . .	1381
9.11.4.134unpack_nas_SLQSSetSysSelectionPrefExt . . . . .	1381
9.11.4.135unpack_nas_SLQSSwiGetHDRPersonality . . . . .	1382
9.11.4.136unpack_nas_SLQSSwiGetHDRProtSubtype . . . . .	1382
9.11.4.137unpack_nas_SLQSSwiGetHRPDStats . . . . .	1382
9.11.4.138unpack_nas_SLQSSwiGetLteCQI . . . . .	1383
9.11.4.139unpack_nas_SLQSSwiGetLteSccRxInfo . . . . .	1383
9.11.4.140unpack_nas_SLQSSwiHDRPersonalityCallback_Ind . . . . .	1383
9.11.4.141unpack_nas_SLQSSwiNetworkDebug . . . . .	1384
9.11.4.142unpack_nas_SLQSSwiPSDetach . . . . .	1384
9.11.4.143unpack_nas_SLQSSwiRandIndicatorCallback_Ind . . . . .	1384
9.11.4.144unpack_valid_nas_GetCDMANetworkParameters . . . . .	1385
9.11.4.145unpack_valid_nas_SLQSGetServingSystem . . . . .	1385
9.11.4.146unpack_valid_nas_SLQSGetSignalStrength . . . . .	1386
9.11.4.147unpack_valid_nas_SLQSNasGetSigInfo . . . . .	1387
9.12 pds.h File Reference . . . . .	1387
9.12.1 Typedef Documentation . . . . .	1390
9.12.1.1 unpack_pds_ForceXTRADownload_t . . . . .	1390
9.12.1.2 unpack_pds_PDSInjectTimeReference_t . . . . .	1390
9.12.1.3 unpack_pds_ResetPDSDData_t . . . . .	1390
9.12.1.4 unpack_pds_SetEventReportCallback_t . . . . .	1390
9.12.1.5 unpack_pds_SetPDSDDefaults_t . . . . .	1390

9.12.1.6	<a href="#">unpack_pds_SetPDSSState_t</a>	1390
9.12.1.7	<a href="#">unpack_pds_SetPortAutomaticTracking_t</a>	1390
9.12.1.8	<a href="#">unpack_pds_SetServiceAutomaticTracking_t</a>	1390
9.12.1.9	<a href="#">unpack_pds_SetXTRAAutomaticDownload_t</a>	1390
9.12.1.10	<a href="#">unpack_pds_SetXTRANetwork_t</a>	1390
9.12.1.11	<a href="#">unpack_pds_SLQSPDSDeterminePosition_t</a>	1390
9.12.1.12	<a href="#">unpack_pds_SLQSPDSInjectAbsoluteTimeReference_t</a>	1390
9.12.1.13	<a href="#">unpack_pds_SLQSPDSInjectPositionData_t</a>	1390
9.12.1.14	<a href="#">unpack_pds_SLQSSetAGPSConfig_t</a>	1390
9.12.1.15	<a href="#">unpack_pds_SLQSSetPositionMethodState_t</a>	1390
9.12.1.16	<a href="#">unpack_pds_StartPDSTrackingSessionExt_t</a>	1390
9.12.1.17	<a href="#">unpack_pds_StopPDSTrackingSession_t</a>	1390
9.12.2	<a href="#">Function Documentation</a>	1390
9.12.2.1	<a href="#">pack_pds_ForceXTRADownload</a>	1391
9.12.2.2	<a href="#">pack_pds_GetPDSDDefaults</a>	1391
9.12.2.3	<a href="#">pack_pds_GetPDSSState</a>	1391
9.12.2.4	<a href="#">pack_pds_GetPortAutomaticTracking</a>	1392
9.12.2.5	<a href="#">pack_pds_GetServiceAutomaticTracking</a>	1392
9.12.2.6	<a href="#">pack_pds_GetXTRAAutomaticDownload</a>	1392
9.12.2.7	<a href="#">pack_pds_GetXTRANetwork</a>	1393
9.12.2.8	<a href="#">pack_pds_GetXTRAVality</a>	1393
9.12.2.9	<a href="#">pack_pds_PDSInjectTimeReference</a>	1393
9.12.2.10	<a href="#">pack_pds_ResetPDSData</a>	1394
9.12.2.11	<a href="#">pack_pds_SetEventReportCallback</a>	1394
9.12.2.12	<a href="#">pack_pds_SetPDSDDefaults</a>	1394
9.12.2.13	<a href="#">pack_pds_SetPDSSState</a>	1395
9.12.2.14	<a href="#">pack_pds_SetPortAutomaticTracking</a>	1395
9.12.2.15	<a href="#">pack_pds_SetServiceAutomaticTracking</a>	1395
9.12.2.16	<a href="#">pack_pds_SetXTRAAutomaticDownload</a>	1396
9.12.2.17	<a href="#">pack_pds_SetXTRANetwork</a>	1396
9.12.2.18	<a href="#">pack_pds_SLQSGetAGPSConfig</a>	1397
9.12.2.19	<a href="#">pack_pds_SLQSGetGPSStateInfo</a>	1397
9.12.2.20	<a href="#">pack_pds_SLQSPDSDeterminePosition</a>	1397
9.12.2.21	<a href="#">pack_pds_SLQSPDSInjectAbsoluteTimeReference</a>	1398
9.12.2.22	<a href="#">pack_pds_SLQSPDSInjectPositionData</a>	1398
9.12.2.23	<a href="#">pack_pds_SLQSSetAGPSConfig</a>	1398
9.12.2.24	<a href="#">pack_pds_SLQSSetPositionMethodState</a>	1399
9.12.2.25	<a href="#">pack_pds_StartPDSTrackingSessionExt</a>	1399
9.12.2.26	<a href="#">pack_pds_StopPDSTrackingSession</a>	1399
9.12.2.27	<a href="#">unpack_pds_ForceXTRADownload</a>	1400



9.12.2.28 unpack_pds_GetPDSDefaults . . . . .	1400
9.12.2.29 unpack_pds_GetPDSSState . . . . .	1400
9.12.2.30 unpack_pds_GetPortAutomaticTracking . . . . .	1401
9.12.2.31 unpack_pds_GetServiceAutomaticTracking . . . . .	1401
9.12.2.32 unpack_pds_GetXTRAAutomaticDownload . . . . .	1402
9.12.2.33 unpack_pds_GetXTRANetwork . . . . .	1402
9.12.2.34 unpack_pds_GetXTRAValidity . . . . .	1402
9.12.2.35 unpack_pds_PDSInjectTimeReference . . . . .	1403
9.12.2.36 unpack_pds_ResetPDSDData . . . . .	1403
9.12.2.37 unpack_pds_SetEventReport_Ind . . . . .	1403
9.12.2.38 unpack_pds_SetEventReportCallback . . . . .	1404
9.12.2.39 unpack_pds_SetPDSDefaults . . . . .	1404
9.12.2.40 unpack_pds_SetPDSSState . . . . .	1404
9.12.2.41 unpack_pds_SetPdsState_Ind . . . . .	1405
9.12.2.42 unpack_pds_SetPortAutomaticTracking . . . . .	1405
9.12.2.43 unpack_pds_SetServiceAutomaticTracking . . . . .	1405
9.12.2.44 unpack_pds_SetXTRAAutomaticDownload . . . . .	1406
9.12.2.45 unpack_pds_SetXTRANetwork . . . . .	1406
9.12.2.46 unpack_pds_SLQSGetAGPSConfig . . . . .	1406
9.12.2.47 unpack_pds_SLQSGetGPSStateInfo . . . . .	1407
9.12.2.48 unpack_pds_SLQSPDSDeterminePosition . . . . .	1407
9.12.2.49 unpack_pds_SLQSPDSInjectAbsoluteTimeReference . . . . .	1407
9.12.2.50 unpack_pds_SLQSPDSInjectPositionData . . . . .	1408
9.12.2.51 unpack_pds_SLQSSetAGPSConfig . . . . .	1408
9.12.2.52 unpack_pds_SLQSSetPositionMethodState . . . . .	1408
9.12.2.53 unpack_pds_StartPDSTrackingSessionExt . . . . .	1409
9.12.2.54 unpack_pds_StopPDSTrackingSession . . . . .	1409
9.13 qaGobiApiTableBandClasses.h File Reference . . . . .	1409
9.13.1 Detailed Description . . . . .	1409
9.13.2 Band Classes (Value - Description) . . . . .	1410
9.13.2.1 LTE Bands . . . . .	1411
9.14 qaGobiApiTableCallControlReturnReasons.h File Reference . . . . .	1413
9.14.1 Detailed Description . . . . .	1413
9.14.2 S1 . . . . .	1413
9.15 qaGobiApiTableCallEndReasons.h File Reference . . . . .	1414
9.15.1 Detailed Description . . . . .	1414
9.15.2 Call end reason codes (Code - Reason) . . . . .	1414
9.15.2.1 Technology-agnostic call end reasons . . . . .	1414
9.15.2.2 EVDO CDMA 1xEV-DO . . . . .	1414
9.15.2.3 WCDMA/GSM call end reasons . . . . .	1415

9.15.2.4	EVDO CDMA 1xEV-DO	1417
9.15.2.5	call end reason type	1417
9.15.2.6	Mobile IP call end reasons (Type=1)	1418
9.15.2.7	Internal call end reasons (Type=2)	1419
9.15.2.8	Call Manager defined call end reasons (Type=3)	1422
9.15.2.9	3GPP specification defined call end reasons (Type=6)	1427
9.15.2.10	PPP call end reasons (Type=7)	1429
9.15.2.11	EHRPD call end reasons (Type=8)	1429
9.15.2.12	IPv6 call end reasons (Type=9)	1430
9.16	qaGobiApiTableCarrierCodes.h File Reference	1430
9.16.1	Detailed Description	1430
9.16.2	Carrier Codes (Number - Carrier)	1430
9.17	qaGobiApiTableCodingScheme.h File Reference	1432
9.17.1	Detailed Description	1432
9.17.2	S1	1432
9.17.2.1	Use of bits 3..0	1432
9.17.3	Coding Group Bits 7..4(0001)	1433
9.17.3.1	use of bits 3..0	1433
9.17.4	Coding Group Bits 7..4(0010)	1433
9.17.4.1	use of bits 3..0	1433
9.17.5	Coding Group Bits 7..4(0011)	1433
9.17.5.1	use of bits 3..0	1433
9.17.6	Coding Group Bits 7..4(01xx)	1433
9.17.6.1	use of bits 3..0	1433
9.17.7	Coding Group Bits 7..4(1001)	1434
9.17.7.1	Reserved coding groups	1434
9.17.8	Coding Group Bits 7..4(1010..1101)	1434
9.17.8.1	Reserved coding groups	1434
9.17.9	Coding Group Bits 7..4(1110)	1434
9.17.9.1	Defined by the WAP Forum	1434
9.17.10	Coding Group Bits 7..4 (1111)	1434
9.17.10.1	Data coding / message handling	1434
9.18	qaGobiApiTableGpsCapabilityCodes.h File Reference	1435
9.18.1	Detailed Description	1435
9.18.2	GPS capability (Value - Capability)	1435
9.19	qaGobiApiTablePowerModes.h File Reference	1435
9.19.1	Detailed Description	1435
9.19.2	Power Modes (Value - Description)	1435
9.20	qaGobiApiTableRadioInterfaces.h File Reference	1436
9.20.1	Detailed Description	1436

9.20.2	Radio interface	1436
9.20.2.1	Technology (Value - Radio Interface Technology)	1436
9.21	qaGobiApiTableRegionCodes.h File Reference	1436
9.21.1	Detailed Description	1437
9.21.2	Region Codes (Code - Region)	1437
9.22	qaGobiApiTableServiceOptions.h File Reference	1437
9.22.1	Detailed Description	1437
9.22.2	Service Option codes (Code - Reason)	1437
9.22.2.1	Description	1437
9.23	qaGobiApiTableSupServiceInfoClasses.h File Reference	1439
9.23.1	Detailed Description	1439
9.23.2	Supplementary Service Information Classes (Value - Service Class)	1440
9.24	qaGobiApiTableSwiAudio.h File Reference	1440
9.24.1	Detailed Description	1440
9.24.2	ACDB Device (Device ID - description)	1440
9.24.3	Physical Interface (Device ID - description - Interface parameters)	1440
9.25	qaGobiApiTableSwiOMADMSessionStatus.h File Reference	1440
9.25.1	Detailed Description	1441
9.25.2	OMA DM Session Status (Session Status - Meaning - Usage)	1441
9.26	qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference	1441
9.26.1	Detailed Description	1441
9.26.2	OMA DM Update Complete Status (Update Complete Status - Meaning - Usage)	1442
9.27	qaGobiApiTableVoiceCallEndReasons.h File Reference	1443
9.27.1	Detailed Description	1443
9.27.2	Voice Call and supplementary services end reason codes (Code - Reason)	1443
9.27.2.1	General	1443
9.27.2.2	service Errors	1445
9.27.2.3	control cause values	1446
9.27.2.4	reject causes	1447
9.27.2.5	reject causes	1448
9.27.2.6	reject causes	1448
9.27.2.7	stratum reject causes	1448
9.27.2.8	reject causes	1449
9.27.2.9	IP end reasons	1449
9.28	qmerrno.h File Reference	1449
9.28.1	Enumeration Type Documentation	1451
9.28.1.1	eQCWWANError	1451
9.28.1.2	qm_wds_ds_profile_extended_err_codes	1456
9.29	qos.h File Reference	1456
9.29.1	Macro Definition Documentation	1458

9.29.1.1	LITEQMI_MAX_QOS_FILTERS	1458
9.29.1.2	LITEQMI_MAX_QOS_FLOW_PER_APN_STATS	1458
9.29.1.3	LITEQMI_MAX_QOS_FLOWS	1458
9.29.2	Function Documentation	1458
9.29.2.1	pack_qos_BindDataPort	1458
9.29.2.2	pack_qos_SLQSQosGetNetworkStatus	1458
9.29.2.3	pack_qos_SLQSQosSwiReadApnExtraParams	1459
9.29.2.4	pack_qos_SLQSQosSwiReadDataStats	1460
9.29.2.5	pack_qos_SLQSSetQosEventCallback	1460
9.29.2.6	unpack_qos_BindDataPort	1461
9.29.2.7	unpack_qos_SLQSQosGetNetworkStatus	1461
9.29.2.8	unpack_qos_SLQSQosSwiReadApnExtraParams	1462
9.29.2.9	unpack_qos_SLQSQosSwiReadDataStats	1462
9.29.2.10	unpack_qos_SLQSSetQosEventCallback	1463
9.29.2.11	unpack_qos_SLQSSetQosEventCallback_ind	1463
9.29.2.12	unpack_qos_SLQSSetQosNWStatusCallback_ind	1464
9.29.2.13	unpack_qos_SLQSSetQosPriEventCallback_ind	1465
9.29.2.14	unpack_qos_SLQSSetQosStatusCallback_ind	1465
9.30	rms.h File Reference	1466
9.30.1	Macro Definition Documentation	1466
9.30.1.1	__LITEQMI_RMS_H__	1466
9.30.2	Function Documentation	1466
9.30.2.1	pack_rms_GetSMSWake	1466
9.30.2.2	pack_rms_SetSMSWake	1467
9.30.2.3	unpack_rms_GetSMSWake	1467
9.30.2.4	unpack_rms_SetSMSWake	1467
9.31	sar.h File Reference	1468
9.31.1	Typedef Documentation	1468
9.31.1.1	unpack_sar_SLQSSetRfSarState_t	1468
9.31.2	Function Documentation	1468
9.31.2.1	pack_sar_SLQSGetRfSarState	1468
9.31.2.2	pack_sar_SLQSSetRfSarState	1469
9.31.2.3	unpack_sar_SLQSGetRfSarState	1469
9.31.2.4	unpack_sar_SLQSSetRfSarState	1469
9.32	sms.h File Reference	1470
9.32.1	Macro Definition Documentation	1473
9.32.1.1	MAX_CDMA_ENC_MO_TXT_MSG_SIZE	1473
9.32.1.2	MAX_MS_TRANSFER_ROUTE_MSG	1473
9.32.1.3	MAX_MSC_ADDRESS_SIZE	1473
9.32.1.4	MAX_MSE_TWS_MSG	1473

9.32.1.5	MAX_SMS_LIST_SIZE	1473
9.32.1.6	MAX_SMS_MESSAGE_SIZE	1473
9.32.1.7	SMS_CONFIG_LEN	1474
9.32.1.8	SMS_MAX_SMS_ROUTES	1474
9.32.1.9	SMS_NUM_OF_SET	1474
9.32.1.10	SMSC_TYPE_LEN	1474
9.32.2	Enumeration Type Documentation	1474
9.32.2.1	eqmiCbkSetStatus	1474
9.32.3	Function Documentation	1474
9.32.3.1	pack_sms_GetSMSCAddress	1474
9.32.3.2	pack_sms_SaveSMS	1474
9.32.3.3	pack_sms_SendSMS	1475
9.32.3.4	pack_sms_SetNewSMSCallback	1475
9.32.3.5	pack_sms_SetSMSCAddress	1475
9.32.3.6	pack_sms_SLQSDeleteSMS	1476
9.32.3.7	pack_sms_SLQSGetIndicationRegister	1476
9.32.3.8	pack_sms_SLQSGetMessageWaiting	1476
9.32.3.9	pack_sms_SLQSGetSMS	1477
9.32.3.10	pack_sms_SLQSGetSmsBroadcastConfig	1477
9.32.3.11	pack_sms_SLQSGetSMSList	1477
9.32.3.12	pack_sms_SLQSGetTransLayerInfo	1478
9.32.3.13	pack_sms_SLQSGetTransNWRegInfo	1478
9.32.3.14	pack_sms_SLQSModifySMSStatus	1478
9.32.3.15	pack_sms_SLQSSendAsyncSMS	1479
9.32.3.16	pack_sms_SLQSSetIndicationRegister	1479
9.32.3.17	pack_sms_SLQSSetSmsBroadcastActivation	1480
9.32.3.18	pack_sms_SLQSSetSmsBroadcastConfig	1480
9.32.3.19	pack_sms_SLQSSetSmsStorage	1480
9.32.3.20	pack_sms_SLQSSmsGetMaxStorageSize	1481
9.32.3.21	pack_sms_SLQSSmsGetMessageProtocol	1481
9.32.3.22	pack_sms_SLQSSmsSetRoutes	1481
9.32.3.23	pack_sms_SLQSSwiGetSMSStorage	1482
9.32.3.24	unpack_sms_GetSMSCAddress	1482
9.32.3.25	unpack_sms_SaveSMS	1482
9.32.3.26	unpack_sms_SendSMS	1483
9.32.3.27	unpack_sms_SetNewSMSCallback	1483
9.32.3.28	unpack_sms_SetNewSMSCallback_ind	1483
9.32.3.29	unpack_sms_SetSMSCAddress	1484
9.32.3.30	unpack_sms_SLQSDeleteSMS	1484
9.32.3.31	unpack_sms_SLQSGetIndicationRegister	1485

9.32.3.32	<a href="#">unpack_sms_SLQSGetMessageWaiting</a>	1485
9.32.3.33	<a href="#">unpack_sms_SLQSGetSMS</a>	1485
9.32.3.34	<a href="#">unpack_sms_SLQSGetSmsBroadcastConfig</a>	1486
9.32.3.35	<a href="#">unpack_sms_SLQSGetSMSList</a>	1486
9.32.3.36	<a href="#">unpack_sms_SLQSGetTransLayerInfo</a>	1486
9.32.3.37	<a href="#">unpack_sms_SLQSGetTransNWRegInfo</a>	1487
9.32.3.38	<a href="#">unpack_sms_SLQSModifySMSStatus</a>	1487
9.32.3.39	<a href="#">unpack_sms_SLQSNWRegInfoCallback_ind</a>	1487
9.32.3.40	<a href="#">unpack_sms_SLQSSendAsyncSMS</a>	1488
9.32.3.41	<a href="#">unpack_sms_SLQSSetIndicationRegister</a>	1488
9.32.3.42	<a href="#">unpack_sms_SLQSSetSmsBroadcastActivation</a>	1488
9.32.3.43	<a href="#">unpack_sms_SLQSSetSmsBroadcastConfig</a>	1489
9.32.3.44	<a href="#">unpack_sms_SLQSSetSmsStorage</a>	1489
9.32.3.45	<a href="#">unpack_sms_SLQSSmsGetMaxStorageSize</a>	1489
9.32.3.46	<a href="#">unpack_sms_SLQSSmsGetMessageProtocol</a>	1490
9.32.3.47	<a href="#">unpack_sms_SLQSSmsSetRoutes</a>	1490
9.32.3.48	<a href="#">unpack_sms_SLQSSwiGetSMSStorage</a>	1491
9.32.3.49	<a href="#">unpack_sms_SLQSTransLayerInfoCallback_ind</a>	1491
9.32.3.50	<a href="#">unpack_sms_SLQSWmsAsyncRawSendCallBack_ind</a>	1491
9.32.3.51	<a href="#">unpack_sms_SLQSWmsMemoryFullCallBack_ind</a>	1492
9.32.3.52	<a href="#">unpack_sms_SLQSWmsMessageWaitingCallBack_ind</a>	1492
9.33	<a href="#">swiaudio.h File Reference</a>	1492
9.33.1	<a href="#">Macro Definition Documentation</a>	1494
9.33.1.1	<a href="#">SWIAUDIO_MAX_LEN_IFACE_TABLE</a>	1494
9.33.2	<a href="#">Typedef Documentation</a>	1494
9.33.2.1	<a href="#">unpack_swiaudio_SLQSSetM2MAudioAVCFG_t</a>	1494
9.33.2.2	<a href="#">unpack_swiaudio_SLQSSetM2MAudioLPBK_t</a>	1494
9.33.2.3	<a href="#">unpack_swiaudio_SLQSSetM2MAudioNVDef_t</a>	1494
9.33.2.4	<a href="#">unpack_swiaudio_SLQSSetM2MAudioProfile_t</a>	1494
9.33.2.5	<a href="#">unpack_swiaudio_SLQSSetM2MAudioVolume_t</a>	1494
9.33.2.6	<a href="#">unpack_swiaudio_SLQSSetM2MAVMute_t</a>	1494
9.33.2.7	<a href="#">unpack_swiaudio_SLQSSetM2MSpkrGain_t</a>	1494
9.33.3	<a href="#">Function Documentation</a>	1494
9.33.3.1	<a href="#">pack_swiaudio_SLQSGetM2MAudioProfile</a>	1494
9.33.3.2	<a href="#">pack_swiaudio_SLQSGetM2MAudioVolume</a>	1495
9.33.3.3	<a href="#">pack_swiaudio_SLQSGetM2MAVMute</a>	1495
9.33.3.4	<a href="#">pack_swiaudio_SLQSGetM2MSpkrGain</a>	1495
9.33.3.5	<a href="#">pack_swiaudio_SLQSSetM2MAudioAVCFG</a>	1496
9.33.3.6	<a href="#">pack_swiaudio_SLQSSetM2MAudioLPBK</a>	1496
9.33.3.7	<a href="#">pack_swiaudio_SLQSSetM2MAudioNVDef</a>	1496

9.33.3.8	<a href="#">pack_swiaudio_SLQSSetM2MAudioProfile</a>	1497
9.33.3.9	<a href="#">pack_swiaudio_SLQSSetM2MAudioVolume</a>	1497
9.33.3.10	<a href="#">pack_swiaudio_SLQSSetM2MAVMute</a>	1498
9.33.3.11	<a href="#">pack_swiaudio_SLQSSetM2MSpkrGain</a>	1498
9.33.3.12	<a href="#">unpack_swiaudio_SLQSGetM2MAudioProfile</a>	1498
9.33.3.13	<a href="#">unpack_swiaudio_SLQSGetM2MAudioVolume</a>	1499
9.33.3.14	<a href="#">unpack_swiaudio_SLQSGetM2MAVMute</a>	1499
9.33.3.15	<a href="#">unpack_swiaudio_SLQSGetM2MSpkrGain</a>	1499
9.33.3.16	<a href="#">unpack_swiaudio_SLQSSetM2MAudioAVCFG</a>	1500
9.33.3.17	<a href="#">unpack_swiaudio_SLQSSetM2MAudioLPBK</a>	1500
9.33.3.18	<a href="#">unpack_swiaudio_SLQSSetM2MAudioNVDef</a>	1500
9.33.3.19	<a href="#">unpack_swiaudio_SLQSSetM2MAudioProfile</a>	1501
9.33.3.20	<a href="#">unpack_swiaudio_SLQSSetM2MAudioVolume</a>	1501
9.33.3.21	<a href="#">unpack_swiaudio_SLQSSetM2MAVMute</a>	1501
9.33.3.22	<a href="#">unpack_swiaudio_SLQSSetM2MSpkrGain</a>	1502
9.34	<a href="#">swiavms.h File Reference</a>	1502
9.34.1	<a href="#">Macro Definition Documentation</a>	1504
9.34.1.1	<a href="#">LITEQMI_MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH</a>	1504
9.34.1.2	<a href="#">LITEQMI_MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH</a>	1504
9.34.1.3	<a href="#">LITEQMI_MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH</a>	1504
9.34.1.4	<a href="#">LITEQMI_MAX_SWIOMA_STR_LEN</a>	1504
9.34.1.5	<a href="#">MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER</a>	1504
9.34.1.6	<a href="#">MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_DESC_LENGTH</a>	1504
9.34.1.7	<a href="#">MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_NAME_LENGTH</a>	1504
9.34.1.8	<a href="#">MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESESSIONINFO_VERSION_LENGTH</a>	1504
9.34.1.9	<a href="#">MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH</a>	1504
9.34.1.10	<a href="#">MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH</a>	1504
9.34.1.11	<a href="#">MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS</a>	1504
9.34.1.12	<a href="#">MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH</a>	1504
9.34.1.13	<a href="#">MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH</a>	1504
9.34.2	<a href="#">Typedef Documentation</a>	1504
9.34.2.1	<a href="#">unpack_swiaavms_SLQSAVMSStopSession_avc2_t</a>	1504
9.34.3	<a href="#">Function Documentation</a>	1504
9.34.3.1	<a href="#">pack_swiaavms_SLQSAVMSGetSettings</a>	1505
9.34.3.2	<a href="#">pack_swiaavms_SLQSAVMSGetSettings_v2</a>	1505
9.34.3.3	<a href="#">pack_swiaavms_SLQSAVMSSendSelection</a>	1506
9.34.3.4	<a href="#">pack_swiaavms_SLQSAVMSSessionGetInfo</a>	1506

9.34.3.5	<a href="#">pack_swiavms_SLQSAVmsSetEventReport</a>	1507
9.34.3.6	<a href="#">pack_swiavms_SLQSAVMSSetSettings</a>	1507
9.34.3.7	<a href="#">pack_swiavms_SLQSAVMSSetSettings_v2</a>	1508
9.34.3.8	<a href="#">pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField</a>	1509
9.34.3.9	<a href="#">pack_swiavms_SLQSAVMSStartSession</a>	1509
9.34.3.10	<a href="#">pack_swiavms_SLQSAVMSStopSession</a>	1510
9.34.3.11	<a href="#">unpack_swiavms_SLQSAVMSEventReportInd</a>	1511
9.34.3.12	<a href="#">unpack_swiavms_SLQSAVMSGetSettings</a>	1511
9.34.3.13	<a href="#">unpack_swiavms_SLQSAVMSGetSettings_v2</a>	1512
9.34.3.14	<a href="#">unpack_swiavms_SLQSAVMSSendSelection</a>	1512
9.34.3.15	<a href="#">unpack_swiavms_SLQSAVMSSessionGetInfo</a>	1512
9.34.3.16	<a href="#">unpack_swiavms_SLQSAVmsSetEventReport</a>	1513
9.34.3.17	<a href="#">unpack_swiavms_SLQSAVMSSetSettings</a>	1513
9.34.3.18	<a href="#">unpack_swiavms_SLQSAVMSSetSettings_v2</a>	1514
9.34.3.19	<a href="#">unpack_swiavms_SLQSAVMSStartSession</a>	1514
9.34.3.20	<a href="#">unpack_swiavms_SLQSAVMSStopSession</a>	1515
9.34.3.21	<a href="#">unpack_swiavms_SLQSAVMSStopSession_avc2</a>	1515
9.35	<a href="#">SwiDataTypes.h File Reference</a>	1516
9.35.1	<a href="#">Detailed Description</a>	1516
9.35.2	<a href="#">Macro Definition Documentation</a>	1516
9.35.2.1	<a href="#">MEMSET_STATIC_OUTPUT_STRUCT</a>	1516
9.35.2.2	<a href="#">QMI_NO_LTE_FW_SUPPORT</a>	1517
9.35.2.3	<a href="#">QMI_TLV_PLACEHOLDER</a>	1517
9.35.2.4	<a href="#">SWI_API</a>	1517
9.35.2.5	<a href="#">UNUSEDPARAM</a>	1517
9.35.3	<a href="#">Typedef Documentation</a>	1517
9.35.3.1	<a href="#">BOOL</a>	1517
9.35.3.2	<a href="#">BYTE</a>	1517
9.35.3.3	<a href="#">CHAR</a>	1517
9.35.3.4	<a href="#">FLOAT</a>	1517
9.35.3.5	<a href="#">INT32</a>	1517
9.35.3.6	<a href="#">INT8</a>	1517
9.35.3.7	<a href="#">LPCSTR</a>	1517
9.35.3.8	<a href="#">qmuint16</a>	1517
9.35.3.9	<a href="#">qmulong</a>	1517
9.35.3.10	<a href="#">SHORT</a>	1517
9.35.3.11	<a href="#">ULONG</a>	1517
9.35.3.12	<a href="#">ULONGLONG</a>	1517
9.35.3.13	<a href="#">USHORT</a>	1517
9.35.3.14	<a href="#">WORD</a>	1517



9.36	swidms.h File Reference	1517
9.36.1	Typedef Documentation	1519
9.36.1.1	unpack_swidms_SLQSSwiDmsSetUsbNetNum_t	1519
9.36.2	Function Documentation	1519
9.36.2.1	pack_swidms_SLQSSwiDmsGetHWWatchdog	1519
9.36.2.2	pack_swidms_SLQSSwiDmsGetMTU	1519
9.36.2.3	pack_swidms_SLQSSwiDmsGetSecureInfo	1519
9.36.2.4	pack_swidms_SLQSSwiDmsGetUsbComp	1520
9.36.2.5	pack_swidms_SLQSSwiDmsGetUsbNetNum	1520
9.36.2.6	pack_swidms_SLQSSwiDmsSetHWWatchdog	1520
9.36.2.7	pack_swidms_SLQSSwiDmsSetMTU	1521
9.36.2.8	pack_swidms_SLQSSwiDmsSetUsbComp	1521
9.36.2.9	pack_swidms_SLQSSwiDmsSetUsbNetNum	1521
9.36.2.10	unpack_swidms_SLQSSwiDmsGetHWWatchdog	1522
9.36.2.11	unpack_swidms_SLQSSwiDmsGetMTU	1522
9.36.2.12	unpack_swidms_SLQSSwiDmsGetSecureInfo	1523
9.36.2.13	unpack_swidms_SLQSSwiDmsGetUsbComp	1523
9.36.2.14	unpack_swidms_SLQSSwiDmsGetUsbNetNum	1523
9.36.2.15	unpack_swidms_SLQSSwiDmsSetHWWatchdog	1524
9.36.2.16	unpack_swidms_SLQSSwiDmsSetMTU	1524
9.36.2.17	unpack_swidms_SLQSSwiDmsSetUsbComp	1524
9.36.2.18	unpack_swidms_SLQSSwiDmsSetUsbNetNum	1525
9.37	swiloc.h File Reference	1525
9.37.1	Typedef Documentation	1525
9.37.1.1	unpack_swiloc_SwiLocSetAutoStart_t	1525
9.37.2	Function Documentation	1526
9.37.2.1	pack_swiloc_SwiLocGetAutoStart	1526
9.37.2.2	pack_swiloc_SwiLocSetAutoStart	1526
9.37.2.3	unpack_swiloc_SwiLocGetAutoStart	1526
9.37.2.4	unpack_swiloc_SwiLocSetAutoStart	1527
9.38	swioma.h File Reference	1527
9.38.1	Macro Definition Documentation	1528
9.38.1.1	LITEQMI_MAX_SWIOMA_STR_LEN	1528
9.38.2	Typedef Documentation	1528
9.38.2.1	unpack_swioma_SLQSOMADMAAlertCallback_t	1528
9.38.2.2	unpack_swioma_SLQSOMADMCancelSession_t	1528
9.38.2.3	unpack_swioma_SLQSOMADMSelectSelection_t	1528
9.38.2.4	unpack_swioma_SLQSOMADMSetSettings_t	1528
9.38.3	Function Documentation	1528
9.38.3.1	pack_swioma_SLQSOMADMAAlertCallback	1528

9.38.3.2	<a href="#">pack_swroma_SLQSOMADMCancelSession</a>	1529
9.38.3.3	<a href="#">pack_swroma_SLQSOMADMGetSessionInfo</a>	1530
9.38.3.4	<a href="#">pack_swroma_SLQSOMADMGetSettings</a>	1530
9.38.3.5	<a href="#">pack_swroma_SLQSOMADMSendSelection</a>	1531
9.38.3.6	<a href="#">pack_swroma_SLQSOMADMSetSettings</a>	1532
9.38.3.7	<a href="#">pack_swroma_SLQSOMADMStartSession</a>	1532
9.38.3.8	<a href="#">unpack_swroma_SLQSOMADMAAlertCallback</a>	1533
9.38.3.9	<a href="#">unpack_swroma_SLQSOMADMAAlertCallback_ind</a>	1533
9.38.3.10	<a href="#">unpack_swroma_SLQSOMADMCancelSession</a>	1534
9.38.3.11	<a href="#">unpack_swroma_SLQSOMADMGetSessionInfo</a>	1534
9.38.3.12	<a href="#">unpack_swroma_SLQSOMADMGetSettings</a>	1535
9.38.3.13	<a href="#">unpack_swroma_SLQSOMADMSendSelection</a>	1535
9.38.3.14	<a href="#">unpack_swroma_SLQSOMADMSetSettings</a>	1536
9.38.3.15	<a href="#">unpack_swroma_SLQSOMADMStartSession</a>	1536
9.39	<a href="#">swromaext.h File Reference</a>	1537
9.39.1	<a href="#">Macro Definition Documentation</a>	1537
9.39.1.1	<a href="#">LITE_SWROMAEXT_MAX_UCS2_DATA_LEN</a>	1537
9.39.1.2	<a href="#">LITEQMI_MAX_SWROMA_STR_LEN</a>	1537
9.39.2	<a href="#">Typedef Documentation</a>	1537
9.39.2.1	<a href="#">unpack_swroma_SLQSOMADMCancelSessionExt_t</a>	1538
9.39.2.2	<a href="#">unpack_swroma_SLQSOMADMSendSelectionExt_t</a>	1538
9.39.2.3	<a href="#">unpack_swroma_SLQSOMADMSetSettingsExt_t</a>	1538
9.39.2.4	<a href="#">unpack_swroma_SLQSOMADMStartSessionExt_t</a>	1538
9.39.3	<a href="#">Function Documentation</a>	1538
9.39.3.1	<a href="#">pack_swroma_SLQSOMADMCancelSessionExt</a>	1538
9.39.3.2	<a href="#">pack_swroma_SLQSOMADMGetSessionInfoExt</a>	1538
9.39.3.3	<a href="#">pack_swroma_SLQSOMADMSendSelectionExt</a>	1539
9.39.3.4	<a href="#">pack_swroma_SLQSOMADMSetSettingsExt</a>	1540
9.39.3.5	<a href="#">pack_swroma_SLQSOMADMStartSessionExt</a>	1540
9.39.3.6	<a href="#">unpack_swroma_SLQSOMADMCancelSessionExt</a>	1541
9.39.3.7	<a href="#">unpack_swroma_SLQSOMADMGetSessionInfoExt</a>	1541
9.39.3.8	<a href="#">unpack_swroma_SLQSOMADMSendSelectionExt</a>	1542
9.39.3.9	<a href="#">unpack_swroma_SLQSOMADMSetSettingsExt</a>	1542
9.39.3.10	<a href="#">unpack_swroma_SLQSOMADMStartSessionExt</a>	1543
9.40	<a href="#">switype_256bit.h File Reference</a>	1543
9.40.1	<a href="#">Macro Definition Documentation</a>	1544
9.40.1.1	<a href="#">SWI_UINT256_BIT_OFFSET</a>	1544
9.40.1.2	<a href="#">SWI_UINT256_BITS_PER_WORD</a>	1544
9.40.1.3	<a href="#">SWI_UINT256_INT_VALUE</a>	1544
9.40.1.4	<a href="#">SWI_UINT256_WORD_COUNT</a>	1544

9.40.1.5	<a href="#">SWI_UINT256_WORD_OFFSET</a>	1544
9.40.2	<a href="#">Typedef Documentation</a>	1544
9.40.2.1	<a href="#">logger</a>	1544
9.40.3	<a href="#">Function Documentation</a>	1544
9.40.3.1	<a href="#">swi_uint256_clear_bit</a>	1544
9.40.3.2	<a href="#">swi_uint256_get_bit</a>	1544
9.40.3.3	<a href="#">swi_uint256_print_mask</a>	1544
9.40.3.4	<a href="#">swi_uint256_set_bit</a>	1544
9.41	<a href="#">SWIWWANCMAPI.h File Reference</a>	1545
9.42	<a href="#">tmd.h File Reference</a>	1545
9.42.1	<a href="#">Macro Definition Documentation</a>	1545
9.42.1.1	<a href="#">MAX_MITIGATION_DEV_ID_LEN</a>	1545
9.42.1.2	<a href="#">MAX_MITIGATION_DEV_LIST_LEN</a>	1545
9.42.1.3	<a href="#">TMD_MAX_DEV_LIST</a>	1546
9.42.2	<a href="#">Function Documentation</a>	1546
9.42.2.1	<a href="#">pack_tmd_SLQSTmdDeRegNotMitigationLvl</a>	1546
9.42.2.2	<a href="#">pack_tmd_SLQSTmdGetMitigationDevList</a>	1546
9.42.2.3	<a href="#">pack_tmd_SLQSTmdGetMitigationLvl</a>	1546
9.42.2.4	<a href="#">pack_tmd_SLQSTmdRegNotMitigationLvl</a>	1547
9.42.2.5	<a href="#">unpack_tmd_SLQSTmdDeRegNotMitigationLvl</a>	1547
9.42.2.6	<a href="#">unpack_tmd_SLQSTmdGetMitigationDevList</a>	1547
9.42.2.7	<a href="#">unpack_tmd_SLQSTmdGetMitigationLvl</a>	1548
9.42.2.8	<a href="#">unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind</a>	1548
9.42.2.9	<a href="#">unpack_tmd_SLQSTmdRegNotMitigationLvl</a>	1549
9.43	<a href="#">uim.h File Reference</a>	1549
9.43.1	<a href="#">Macro Definition Documentation</a>	1553
9.43.1.1	<a href="#">MAX_ATR_LENGTH</a>	1553
9.43.1.2	<a href="#">MAX_DESCRIPTION_LENGTH</a>	1553
9.43.1.3	<a href="#">MAX_ICCID_LENGTH</a>	1553
9.43.1.4	<a href="#">MAX_NO_OF_APPLICATIONS</a>	1553
9.43.1.5	<a href="#">MAX_NO_OF_SLOTS</a>	1553
9.43.1.6	<a href="#">MAX_PHY_SLOTS_INFO</a>	1553
9.43.1.7	<a href="#">MAX_SLOTS_STATUS</a>	1553
9.43.1.8	<a href="#">UIM_MAX_ACTIVE_PERS_FEATURES</a>	1553
9.43.1.9	<a href="#">UIM_MAX_CONTENT_LENGTH</a>	1553
9.43.1.10	<a href="#">UIM_MAX_DESCRIPTION_LENGTH</a>	1553
9.43.1.11	<a href="#">UIM_MAX_NO_OF_APPLICATIONS</a>	1553
9.43.1.12	<a href="#">UIM_MAX_NO_OF_SLOTS</a>	1553
9.43.1.13	<a href="#">UIM_UINT8_MAX_STRING_SZ</a>	1553
9.43.2	<a href="#">Typedef Documentation</a>	1553

9.43.2.1	<a href="#">unpack_uim_SLQSUIMPowerDown_t</a>	1553
9.43.2.2	<a href="#">unpack_uim_SLQSUIMPowerUp_t</a>	1554
9.43.2.3	<a href="#">unpack_uim_SLQSUIMRefreshComplete_t</a>	1554
9.43.2.4	<a href="#">unpack_uim_SLQSUIMRefreshOK_t</a>	1554
9.43.2.5	<a href="#">unpack_uim_SLQSUIMRefreshRegister_t</a>	1554
9.43.2.6	<a href="#">unpack_uim_SLQSUIMReset_t</a>	1554
9.43.2.7	<a href="#">unpack_uim_SLQSUIMSetServiceStatus_t</a>	1554
9.43.2.8	<a href="#">unpack_uim_SLQSUIMSwitchSlot_t</a>	1554
9.43.3	<a href="#">Function Documentation</a>	1554
9.43.3.1	<a href="#">pack_uim_ChangePin</a>	1554
9.43.3.2	<a href="#">pack_uim_GetCardStatus</a>	1554
9.43.3.3	<a href="#">pack_uim_ReadTransparent</a>	1555
9.43.3.4	<a href="#">pack_uim_SetPinProtection</a>	1555
9.43.3.5	<a href="#">pack_uim_SLQSUIMAuthenticate</a>	1555
9.43.3.6	<a href="#">pack_uim_SLQSUIDepersonalization</a>	1556
9.43.3.7	<a href="#">pack_uim_SLQSUIEventRegister</a>	1556
9.43.3.8	<a href="#">pack_uim_SLQSUIGetConfiguration</a>	1556
9.43.3.9	<a href="#">pack_uim_SLQSUIGetFileAttributes</a>	1557
9.43.3.10	<a href="#">pack_uim_SLQSUIGetServiceStatus</a>	1557
9.43.3.11	<a href="#">pack_uim_SLQSUIGetSlotsStatus</a>	1558
9.43.3.12	<a href="#">pack_uim_SLQSUIPowerDown</a>	1558
9.43.3.13	<a href="#">pack_uim_SLQSUIPowerUp</a>	1558
9.43.3.14	<a href="#">pack_uim_SLQSUIReadRecord</a>	1559
9.43.3.15	<a href="#">pack_uim_SLQSUIRefreshComplete</a>	1559
9.43.3.16	<a href="#">pack_uim_SLQSUIRefreshGetLastEvent</a>	1559
9.43.3.17	<a href="#">pack_uim_SLQSUIRefreshOK</a>	1560
9.43.3.18	<a href="#">pack_uim_SLQSUIRefreshRegister</a>	1560
9.43.3.19	<a href="#">pack_uim_SLQSUIReset</a>	1560
9.43.3.20	<a href="#">pack_uim_SLQSUISetServiceStatus</a>	1561
9.43.3.21	<a href="#">pack_uim_SLQSUISwitchSlot</a>	1561
9.43.3.22	<a href="#">pack_uim_SLQSUIWriteRecord</a>	1562
9.43.3.23	<a href="#">pack_uim_SLQSUIWriteTransparent</a>	1562
9.43.3.24	<a href="#">pack_uim_UnblockPin</a>	1562
9.43.3.25	<a href="#">pack_uim_VerifyPin</a>	1563
9.43.3.26	<a href="#">unpack_uim_ChangePin</a>	1563
9.43.3.27	<a href="#">unpack_uim_GetCardStatus</a>	1563
9.43.3.28	<a href="#">unpack_uim_GetCardStatusV2</a>	1564
9.43.3.29	<a href="#">unpack_uim_ReadTransparent</a>	1564
9.43.3.30	<a href="#">unpack_uim_SetPinProtection</a>	1564
9.43.3.31	<a href="#">unpack_uim_SetUimSlotStatusChangeCallback_ind</a>	1565

9.43.3.32 unpack_uim_SLQSUIMAuthenticate . . . . .	1565
9.43.3.33 unpack_uim_SLQSUIMDepersonalization . . . . .	1565
9.43.3.34 unpack_uim_SLQSUIMEventRegister . . . . .	1566
9.43.3.35 unpack_uim_SLQSUIMGetConfiguration . . . . .	1566
9.43.3.36 unpack_uim_SLQSUIMGetFileAttributes . . . . .	1567
9.43.3.37 unpack_uim_SLQSUIMGetServiceStatus . . . . .	1567
9.43.3.38 unpack_uim_SLQSUIMGetSlotsStatus . . . . .	1567
9.43.3.39 unpack_uim_SLQSUIMGetSlotsStatusV2 . . . . .	1568
9.43.3.40 unpack_uim_SLQSUIMPowerDown . . . . .	1568
9.43.3.41 unpack_uim_SLQSUIMPowerUp . . . . .	1568
9.43.3.42 unpack_uim_SLQSUIMReadRecord . . . . .	1569
9.43.3.43 unpack_uim_SLQSUIMRefreshCallback_Ind . . . . .	1569
9.43.3.44 unpack_uim_SLQSUIMRefreshComplete . . . . .	1569
9.43.3.45 unpack_uim_SLQSUIMRefreshGetLastEvent . . . . .	1570
9.43.3.46 unpack_uim_SLQSUIMRefreshOK . . . . .	1570
9.43.3.47 unpack_uim_SLQSUIMRefreshRegister . . . . .	1570
9.43.3.48 unpack_uim_SLQSUIMReset . . . . .	1571
9.43.3.49 unpack_uim_SLQSUIMSetServiceStatus . . . . .	1571
9.43.3.50 unpack_uim_SLQSUIMSetStatusChangeCallBack_ind . . . . .	1571
9.43.3.51 unpack_uim_SLQSUIMSwitchSlot . . . . .	1572
9.43.3.52 unpack_uim_SLQSUIMWriteRecord . . . . .	1572
9.43.3.53 unpack_uim_SLQSUIMWriteTransparent . . . . .	1572
9.43.3.54 unpack_uim_UnblockPin . . . . .	1573
9.43.3.55 unpack_uim_UnblockPinV2 . . . . .	1573
9.43.3.56 unpack_uim_VerifyPin . . . . .	1573
9.44 voice.h File Reference . . . . .	1574
9.44.1 Detailed Description . . . . .	1579
9.44.2 S1 . . . . .	1579
9.44.3 Macro Definition Documentation . . . . .	1579
9.44.3.1 BARRING_PASSWORD_LENGTH . . . . .	1579
9.44.3.2 MAX_VOICE_CALL_NO_LEN . . . . .	1579
9.44.3.3 MAX_VOICE_DESCRIPTION_LENGTH . . . . .	1579
9.44.3.4 MAXVOICEUSSDLENGTH . . . . .	1579
9.44.3.5 VOICE_MAX_NO_OF_CALLS . . . . .	1580
9.44.4 Typedef Documentation . . . . .	1580
9.44.4.1 unpack_voice_AnswerUSSD_t . . . . .	1580
9.44.4.2 unpack_voice_CancelUSSD_t . . . . .	1580
9.44.4.3 unpack_voice_OriginateUSSD_t . . . . .	1580
9.44.4.4 unpack_voice_SLQSVoiceALSSelectLine_t . . . . .	1580
9.44.4.5 unpack_voice_SLQSVoiceALSSetLineSwitching_t . . . . .	1580

9.44.4.6	unpack_voice_SLQSVoiceBindSubscription_t	1580
9.44.4.7	unpack_voice_SLQSVoiceIndicationRegister_t	1580
9.44.4.8	unpack_voice_SLQSVoiceOrigUSSDNoWait_t	1580
9.44.4.9	unpack_voice_SLQSVoiceSetPreferredPrivacy_t	1580
9.44.5	Enumeration Type Documentation	1580
9.44.5.1	liteServiceClassInformation	1580
9.44.6	Function Documentation	1580
9.44.6.1	pack_voice_AnswerUSSD	1580
9.44.6.2	pack_voice_CancelUSSD	1581
9.44.6.3	pack_voice_OriginateUSSD	1581
9.44.6.4	pack_voice_SLQSVoiceOriginateUSSD	1581
9.44.6.5	pack_voice_SLQSVoiceALSSelectLine	1582
9.44.6.6	pack_voice_SLQSVoiceALSSetLineSwitching	1582
9.44.6.7	pack_voice_SLQSVoiceAnswerCall	1583
9.44.6.8	pack_voice_SLQSVoiceBindSubscription	1583
9.44.6.9	pack_voice_SLQSVoiceBurstDTMF	1583
9.44.6.10	pack_voice_SLQSVoiceDialCall	1584
9.44.6.11	pack_voice_SLQSVoiceEndCall	1584
9.44.6.12	pack_voice_SLQSVoiceGetAllCallInfo	1584
9.44.6.13	pack_voice_SLQSVoiceGetCallBarring	1585
9.44.6.14	pack_voice_SLQSVoiceGetCallForwardingStatus	1585
9.44.6.15	pack_voice_SLQSVoiceGetCallInfo	1586
9.44.6.16	pack_voice_SLQSVoiceGetCallWaiting	1586
9.44.6.17	pack_voice_SLQSVoiceGetCLIP	1586
9.44.6.18	pack_voice_SLQSVoiceGetCLIR	1587
9.44.6.19	pack_voice_SLQSVoiceGetCNAP	1587
9.44.6.20	pack_voice_SLQSVoiceGetCOLP	1587
9.44.6.21	pack_voice_SLQSVoiceGetCOLR	1588
9.44.6.22	pack_voice_SLQSVoiceGetConfig	1588
9.44.6.23	pack_voice_SLQSVoiceIndicationRegister	1588
9.44.6.24	pack_voice_SLQSVoiceManageCalls	1589
9.44.6.25	pack_voice_SLQSVoiceOrigUSSDNoWait	1589
9.44.6.26	pack_voice_SLQSVoiceSendFlash	1589
9.44.6.27	pack_voice_SLQSVoiceSetCallBarringPassword	1590
9.44.6.28	pack_voice_SLQSVoiceSetConfig	1590
9.44.6.29	pack_voice_SLQSVoiceSetPreferredPrivacy	1591
9.44.6.30	pack_voice_SLQSVoiceSetSUPSService	1591
9.44.6.31	pack_voice_SLQSVoiceStartContDTMF	1591
9.44.6.32	pack_voice_SLQSVoiceStopContDTMF	1592
9.44.6.33	unpack_voice_allCallStatusCallback_ind	1592

9.44.6.34 unpack_voice_AnswerUSSD . . . . .	1592
9.44.6.35 unpack_voice_CancelUSSD . . . . .	1593
9.44.6.36 unpack_voice_DTMFEventCallback_ind . . . . .	1593
9.44.6.37 unpack_voice_OriginateUSSD . . . . .	1593
9.44.6.38 unpack_voice_OTASPStatusCallback_ind . . . . .	1594
9.44.6.39 unpack_voice_SLQSOriginateUSSD . . . . .	1594
9.44.6.40 unpack_voice_SLQSVoiceALSSelectLine . . . . .	1595
9.44.6.41 unpack_voice_SLQSVoiceALSSetLineSwitching . . . . .	1595
9.44.6.42 unpack_voice_SLQSVoiceAnswerCall . . . . .	1595
9.44.6.43 unpack_voice_SLQSVoiceBindSubscription . . . . .	1596
9.44.6.44 unpack_voice_SLQSVoiceBurstDTMF . . . . .	1596
9.44.6.45 unpack_voice_SLQSVoiceDialCall . . . . .	1596
9.44.6.46 unpack_voice_SLQSVoiceEndCall . . . . .	1597
9.44.6.47 unpack_voice_SLQSVoiceGetAllCallInfo . . . . .	1597
9.44.6.48 unpack_voice_SLQSVoiceGetCallBarring . . . . .	1597
9.44.6.49 unpack_voice_SLQSVoiceGetCallForwardingStatus . . . . .	1598
9.44.6.50 unpack_voice_SLQSVoiceGetCallInfo . . . . .	1598
9.44.6.51 unpack_voice_SLQSVoiceGetCallWaiting . . . . .	1598
9.44.6.52 unpack_voice_SLQSVoiceGetCLIP . . . . .	1599
9.44.6.53 unpack_voice_SLQSVoiceGetCLIR . . . . .	1599
9.44.6.54 unpack_voice_SLQSVoiceGetCNAP . . . . .	1599
9.44.6.55 unpack_voice_SLQSVoiceGetCOLP . . . . .	1600
9.44.6.56 unpack_voice_SLQSVoiceGetCOLR . . . . .	1600
9.44.6.57 unpack_voice_SLQSVoiceGetConfig . . . . .	1600
9.44.6.58 unpack_voice_SLQSVoiceIndicationRegister . . . . .	1601
9.44.6.59 unpack_voice_SLQSVoiceManageCalls . . . . .	1601
9.44.6.60 unpack_voice_SLQSVoiceOrigUSSDNoWait . . . . .	1601
9.44.6.61 unpack_voice_SLQSVoiceSendFlash . . . . .	1602
9.44.6.62 unpack_voice_SLQSVoiceSetCallBarringPassword . . . . .	1602
9.44.6.63 unpack_voice_SLQSVoiceSetConfig . . . . .	1602
9.44.6.64 unpack_voice_SLQSVoiceSetPreferredPrivacy . . . . .	1603
9.44.6.65 unpack_voice_SLQSVoiceSetSUPSService . . . . .	1603
9.44.6.66 unpack_voice_SLQSVoiceStartContDTMF . . . . .	1603
9.44.6.67 unpack_voice_SLQSVoiceStopContDTMF . . . . .	1604
9.44.6.68 unpack_voice_SLQSVoiceSUPSCallback_ind . . . . .	1604
9.44.6.69 unpack_voice_SUPSNotificationCallback_ind . . . . .	1604
9.44.6.70 unpack_voice_USSDNotificationCallback_ind . . . . .	1605
9.44.6.71 unpack_voice_VoiceInfoRecCallback_ind . . . . .	1605
9.44.6.72 unpack_voice_voicePrivacyChangeCallback_ind . . . . .	1606
9.45 wds.h File Reference . . . . .	1606

9.45.1	Detailed Description	1613
9.45.2	SO Mask	1613
9.45.3	RAT Mask	1614
9.45.4	Macro Definition Documentation	1614
9.45.4.1	BYT_STAT_STAT_MASK	1614
9.45.4.2	IPV6_ADDRESS_ARRAY_SIZE	1614
9.45.4.3	LITE_MAX_PCOID_LIST	1614
9.45.4.4	LITE_MAX_PDN_THROTTLE_TIMER	1614
9.45.4.5	MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE	1615
9.45.4.6	MAX_WDS_NAME_ARRAY_SIZE	1615
9.45.4.7	PACK_WDS_IPV4	1615
9.45.4.8	PACK_WDS_IPV6	1615
9.45.4.9	WDS_DHCP_MAX_NUM_OPTIONS	1615
9.45.4.10	WDS_DHCP_OPTION_DATA_BUF_SIZE	1615
9.45.4.11	WDS_PROFILE_3GPP	1615
9.45.4.12	WDS_PROFILE_3GPP2	1615
9.45.4.13	WDS_TFTID_SOURCE_IP_SIZE	1615
9.45.5	Typedef Documentation	1615
9.45.5.1	unpack_wds_DHCPv4ClientLeaseChange_t	1615
9.45.5.2	unpack_wds_RMTransferStatistics_ind_t	1615
9.45.5.3	unpack_wds_SetAutoconnect_t	1615
9.45.5.4	unpack_wds_SetDefaultProfile_t	1615
9.45.5.5	unpack_wds_SetDefaultProfileNum_t	1615
9.45.5.6	unpack_wds_SetMobileIP_t	1615
9.45.5.7	unpack_wds_SetMobileIPParameters_t	1615
9.45.5.8	unpack_wds_SetMuxID_t	1615
9.45.5.9	unpack_wds_SLQSResetPacketStatics_t	1615
9.45.5.10	unpack_wds_SLQSSet3GPPConfigItem_t	1615
9.45.5.11	unpack_wds_SLQSSetWdsEventCallback_t	1615
9.45.5.12	unpack_wds_SLQSSetDHCPv4ClientConfig_t	1615
9.45.5.13	unpack_wds_SLQSSetLoopback_t	1615
9.45.5.14	unpack_wds_SLQSStopDataSession_t	1615
9.45.5.15	unpack_wds_SLQSSwiProfileChangeCallback_t	1615
9.45.5.16	unpack_wds_SLQSWdsGoActive_t	1615
9.45.5.17	unpack_wds_SLQSWdsGoDormant_t	1615
9.45.5.18	unpack_wds_SLQSWdsSetEventReport_t	1615
9.45.5.19	UnpackQmiProfileInfo	1616
9.45.5.20	UnpackQmiProfileInfoV2	1616
9.45.6	Enumeration Type Documentation	1616
9.45.6.1	lteQmiDataBearerMasks	1616



9.45.7	Function Documentation	1616
9.45.7.1	pack_wds_DHCPv4ClientLeaseChange	1616
9.45.7.2	pack_wds_GetAutoconnect	1617
9.45.7.3	pack_wds_GetByteTotals	1617
9.45.7.4	pack_wds_GetConnectionRate	1618
9.45.7.5	pack_wds_GetDataBearerTechnology	1618
9.45.7.6	pack_wds_GetDefaultProfile	1618
9.45.7.7	pack_wds_GetDefaultProfileNum	1619
9.45.7.8	pack_wds_GetDefaultProfileV2	1619
9.45.7.9	pack_wds_GetDormancyState	1620
9.45.7.10	pack_wds_GetLastMobileIPError	1620
9.45.7.11	pack_wds_GetMobileIP	1620
9.45.7.12	pack_wds_GetMobileIPProfile	1621
9.45.7.13	pack_wds_GetPacketStatistics	1621
9.45.7.14	pack_wds_GetPacketStatus	1622
9.45.7.15	pack_wds_GetSessionDuration	1622
9.45.7.16	pack_wds_GetSessionDurationV2	1623
9.45.7.17	pack_wds_GetSessionState	1623
9.45.7.18	pack_wds_RMSetTransferStatistics	1623
9.45.7.19	pack_wds_SetAutoconnect	1624
9.45.7.20	pack_wds_SetDefaultProfile	1624
9.45.7.21	pack_wds_SetDefaultProfileNum	1625
9.45.7.22	pack_wds_SetMobileIP	1625
9.45.7.23	pack_wds_SetMobileIPParameters	1625
9.45.7.24	pack_wds_SetMobileIPProfile	1626
9.45.7.25	pack_wds_SetMuxID	1626
9.45.7.26	pack_wds_SLQSCreateProfile	1627
9.45.7.27	pack_wds_SLQSDeleteProfile	1627
9.45.7.28	pack_wds_SLQSGet3GPPConfigItem	1627
9.45.7.29	pack_wds_SLQSGetCurrDataSystemStat	1628
9.45.7.30	pack_wds_SLQSGetCurrentChannelRate	1628
9.45.7.31	pack_wds_SLQSGetDataBearerTechnology	1629
9.45.7.32	pack_wds_SLQSGetDUNCallInfo	1629
9.45.7.33	pack_wds_SLQSGetProfileSettings	1630
9.45.7.34	pack_wds_SLQSGetProfileSettingsV2	1630
9.45.7.35	pack_wds_SLQSGetRuntimeSettings	1631
9.45.7.36	pack_wds_SLQSModifyProfile	1631
9.45.7.37	pack_wds_SLQSResetPacketStatics	1631
9.45.7.38	pack_wds_SLQSSet3GPPConfigItem	1632
9.45.7.39	pack_wds_SLQSSetIPFamilyPreference	1632

9.45.7.40 pack_wds_SLQSSetWdsEventCallback . . . . .	1633
9.45.7.41 pack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1633
9.45.7.42 pack_wds_SLQSSetLoopback . . . . .	1633
9.45.7.43 pack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1634
9.45.7.44 pack_wds_SLQSSetLoopback . . . . .	1634
9.45.7.45 pack_wds_SLQSStartDataSession . . . . .	1635
9.45.7.46 pack_wds_SLQSStopDataSession . . . . .	1635
9.45.7.47 pack_wds_SLQSSwiProfileChangeCallback . . . . .	1635
9.45.7.48 pack_wds_SLQSWdsGoActive . . . . .	1636
9.45.7.49 pack_wds_SLQSWdsGoDormant . . . . .	1636
9.45.7.50 pack_wds_SLQSWdsSetEventReport . . . . .	1637
9.45.7.51 pack_wds_SLQSWdsSwiPDPRuntimeSettings . . . . .	1637
9.45.7.52 unpack_wds_DHCPv4ClientLease_ind . . . . .	1638
9.45.7.53 unpack_wds_DHCPv4ClientLeaseChange . . . . .	1638
9.45.7.54 unpack_wds_GetAutoconnect . . . . .	1638
9.45.7.55 unpack_wds_GetByteTotals . . . . .	1639
9.45.7.56 unpack_wds_GetConnectionRate . . . . .	1639
9.45.7.57 unpack_wds_GetDataBearerTechnology . . . . .	1639
9.45.7.58 unpack_wds_GetDefaultProfile . . . . .	1640
9.45.7.59 unpack_wds_GetDefaultProfileNum . . . . .	1640
9.45.7.60 unpack_wds_GetDefaultProfileV2 . . . . .	1640
9.45.7.61 unpack_wds_GetDormancyState . . . . .	1641
9.45.7.62 unpack_wds_GetLastMobileIPError . . . . .	1641
9.45.7.63 unpack_wds_GetMobileIP . . . . .	1641
9.45.7.64 unpack_wds_GetMobileIPProfile . . . . .	1642
9.45.7.65 unpack_wds_GetPacketStatistics . . . . .	1642
9.45.7.66 unpack_wds_GetPacketStatus . . . . .	1642
9.45.7.67 unpack_wds_GetSessionDuration . . . . .	1643
9.45.7.68 unpack_wds_GetSessionDurationV2 . . . . .	1643
9.45.7.69 unpack_wds_GetSessionState . . . . .	1643
9.45.7.70 unpack_wds_RMSetTransferStatistics . . . . .	1644
9.45.7.71 unpack_wds_RMTransferStatistics_ind . . . . .	1644
9.45.7.72 unpack_wds_SetAutoconnect . . . . .	1644
9.45.7.73 unpack_wds_SetDefaultProfile . . . . .	1645
9.45.7.74 unpack_wds_SetDefaultProfileNum . . . . .	1645
9.45.7.75 unpack_wds_SetMobileIP . . . . .	1645
9.45.7.76 unpack_wds_SetMobileIPParameters . . . . .	1646
9.45.7.77 unpack_wds_SetMobileIPProfile . . . . .	1646
9.45.7.78 unpack_wds_SetMuxID . . . . .	1646
9.45.7.79 unpack_wds_SLQSCreateProfile . . . . .	1647

9.45.7.80 unpack_wds_SLQSDeleteProfile . . . . .	1647
9.45.7.81 unpack_wds_SLQSDUNCallInfoCallBack_ind . . . . .	1647
9.45.7.82 unpack_wds_SLQSGet3GPPConfigItem . . . . .	1648
9.45.7.83 unpack_wds_SLQSGetCurrDataSystemStat . . . . .	1648
9.45.7.84 unpack_wds_SLQSGetCurrentChannelRate . . . . .	1648
9.45.7.85 unpack_wds_SLQSGetDataBearerTechnology . . . . .	1649
9.45.7.86 unpack_wds_SLQSGetDUNCallInfo . . . . .	1649
9.45.7.87 unpack_wds_SLQSGetProfileSettings . . . . .	1649
9.45.7.88 unpack_wds_SLQSGetProfileSettingsV2 . . . . .	1650
9.45.7.89 unpack_wds_SLQSGetRuntimeSettings . . . . .	1650
9.45.7.90 unpack_wds_SLQSModifyProfile . . . . .	1650
9.45.7.91 unpack_wds_SLQSResetPacketStatics . . . . .	1651
9.45.7.92 unpack_wds_SLQSSet3GPPConfigItem . . . . .	1651
9.45.7.93 unpack_wds_SLQSSetIPFamilyPreference . . . . .	1651
9.45.7.94 unpack_wds_SLQSSetPacketSrvStatusCallback . . . . .	1652
9.45.7.95 unpack_wds_SLQSSetWdsEventCallback . . . . .	1652
9.45.7.96 unpack_wds_SLQSSetWdsEventCallback_ind . . . . .	1652
9.45.7.97 unpack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1653
9.45.7.98 unpack_wds_SLQSSetLoopback . . . . .	1653
9.45.7.99 unpack_wds_SLQSSetDHCPv4ClientConfig . . . . .	1653
9.45.7.100unpack_wds_SLQSSetLoopback . . . . .	1654
9.45.7.101unpack_wds_SLQSSetDataSession . . . . .	1654
9.45.7.102unpack_wds_SLQSSetDataSession . . . . .	1654
9.45.7.103unpack_wds_SLQSSetProfileChangeCallback . . . . .	1655
9.45.7.104unpack_wds_SLQSSetProfileChangeCallback_Ind . . . . .	1655
9.45.7.105unpack_wds_SLQSWdsGoActive . . . . .	1655
9.45.7.106unpack_wds_SLQSWdsGoDormant . . . . .	1656
9.45.7.107unpack_wds_SLQSWdsSetEventReport . . . . .	1656
9.45.7.108unpack_wds_SLQSWdsSetPDPRuntimeSettings . . . . .	1657



# 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:

QMI pack/unpack (lite-qmi) . . . . .	<a href="#">31</a>
Streaming Download Protocol (lite-fw) . . . . .	<a href="#">32</a>





## Chapter 3

# Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Tables</a> . . . . .	33
----------------------------------	----



## Chapter 4

# Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

_litewf_FirmwareFileInfo	35
_litewf_FirmwareInfo	36
_litewf_FirmwarePartNo	37
altSrcInfo_t	38
appStats	39
audio_RXAGCList	42
audio_RXAVCList	43
audio_RXPCMIIRFtr	43
audio_TXAGCList	45
audio_TXPCMIIRFtr	46
CarrierImage_t	47
cat_AlPhaIdentifierTlv	49
cat_commonEventTlv	49
cat_currentCatEvent	50
cat_EndProactiveSessionTlv	51
cat_EventIDDDataTlv	51
cat_EventListTlv	51
cat_RefreshTlv	52
cdmaSSInfo	52
connectionStatus	53
crashInfoParams	54
crashInformation	54
currNetworkInfo	55
dms_ActivationStatusTlv	56
dms_devCaps	56
dms_devCurSubsCaps	58
dms_devMaxCfgListCaps	58
dms_devMaxSubsCaps	59
dms_devMultiSimCaps	60
dms_devMultiSimVoiceDataCaps	61
dms_devSubsCfgList	61
dms_devSubsFeatureModeCaps	62
dms_devSubsList	63
dms_devSubsVoiceDataCaps	63
dms_devSubsVoiceDataList	64
dms_LteBandsSupport	64
dms_OperatingModeTlv	65
dms_PSMActiveTimerIndTlv	66

dms_PSMActiveTimerTlv	66
dms_PSMDurationDueToOOSTlv	67
dms_PSMDurationThresholdTlv	67
dms_PSMEarlyWakeupTimeTlv	68
dms_PSMEnableStateIndTlv	69
dms_PSMEnableStateTlv	69
dms_PSMPeriodicUpdateTimerIndTlv	70
dms_PSMPeriodicUpdateTimerTlv	70
dms_PSMRandomizationWindowTlv	71
dms_TemperatureTlv	71
dms_UimAutoSwitchActSlotTlv	72
dms_UimStatusTlv	73
dms_VoltageTlv	73
DMScustSettingInfo	74
DMScustSettingList	75
DMSgetCustomFeatureV2	76
DMSgetCustomInput	76
dunchannelRate	77
eriDataparams	77
eTWSPLMNInfoTlv	78
FMSImageElement	78
FMSImageIdElement	79
FMSImageIDEntries	80
FMSImageList	81
FMSPrefImageList	82
hdrSSInfo	82
image_info_t	83
ims_AMRModelInfo	84
ims_AMROctAlgnInfo	85
ims_AMRWBModelInfo	85
ims_AMRWBOctAlgnInfo	86
ims_CSCFPortNameInfo	86
ims_EnabAMRWBInfo	87
ims_EnabSCRAMRInfo	87
ims_EnabSCRAMRWBInfo	88
ims_IMSDomainInfo	88
ims_IMSTestModelInfo	89
ims_MinSessExpInfo	89
ims_PCSCFPortInfo	90
ims_PhCtxtURIInfo	90
ims_RngBkTmrInfo	91
ims_RngTmrInfo	91
ims_RTPRTCPInactTmrDurInfo	92
ims_SessDurInfo	92
ims_SigCompEnInfo	93
ims_SIPPortInfo	93
ims_SIPRegnTmrInfo	94
ims_SMSFmtInfo	94
ims_SMSolPNwInfo	95
ims_SubscrTmrInfo	95
ims_TmrT1Info	95
ims_TmrT2Info	96
ims_TmrTflInfo	96
imsa_IMSFailErrCodeTlv	97
imsa_IMSRegStatusErrorCodeInfo	97
imsa_IMSRegStatusInfo	98
imsa_NewIMSRegStatusInfo	99
imsa_RatHandoverStatusInfo	99

imsa_SmsRatInfo	100
imsa_SmsSvcStatusInfo	101
imsa_UtRatInfo	101
imsa_UtSvcStatusInfo	102
imsa_VoipRatInfo	102
imsa_VoipSvcStatusInfo	103
imsa_VtRatInfo	103
imsa_VtSvcStatusInfo	104
ipv6AddressInfo	104
LibPackGPRSRequestedQoS	105
LibPackPCOIDList	105
LibPackPDNThrottleTimer	106
LibpackProfile3GPP	106
LibpackProfile3GPP2	113
LibpackProfile3GPPV2	119
LibPackprofile_3GPP	129
LibPackprofile_3GPP2	135
LibPackProfileMnc	140
LibPackQosClassID	141
LibPackTFTIDParams	142
LibPackUMTSQoS	144
LibPackUMTSReqQoSSigInd	147
loc_accelAcceptReady	147
loc_accelTempAcceptReady	148
loc_AppProviderInfoTlv	149
loc_BdsSV	150
loc_BdsSVInfo	151
loc_CellDb	151
loc_ClkInfo	152
loc_FixCriteriaStatusTlv	153
loc_GnssData	154
loc_gpsTime	155
loc_gyroAcceptReady	156
loc_gyroTempAcceptReady	157
loc_HorAccuracyLvlTlv	158
loc_IntermediateRptStateTlv	158
loc_IPv4Config	159
loc_IPv4Info	159
loc_IPv6Config	160
loc_IPv6Info	160
loc_LocApplicationInfo	161
loc_MinIntervalTlv	162
loc_precisionDilution	163
loc_satelliteInfo	163
loc_sensorDataUsage	166
loc_SV	166
loc_SVInfo	167
loc_svUsedforFix	168
loc_urlAddr	168
loc_URLAddrInfo	169
lteSSInfo	169
messageModeTlv	170
nas_acqOrderPref	171
nas_AcqOrderPrefTlv	171
nas_ActPilotPNElement	172
nas_AddCDMASysInfo	173
nas_AddSysInfo	173
nas_BandPrefInfoTlv	174

<a href="#">nas_BandPrefTlv</a>	175
<a href="#">nas_CallBarringSysInfo</a>	177
<a href="#">nas_callBarStatus</a>	177
<a href="#">nas_CDMAChannel</a>	178
<a href="#">nas_CDMAECIOThresh</a>	179
<a href="#">nas_CDMAInfo</a>	180
<a href="#">nas_CDMARSSIThresh</a>	181
<a href="#">nas_CDMASysInfo</a>	181
<a href="#">nas_CDMASysInfoExt</a>	185
<a href="#">nas_cellParams</a>	185
<a href="#">nas_ciotAcqOrderPref</a>	186
<a href="#">nas_CiotAcqOrderPrefTlv</a>	187
<a href="#">nas_CiotLteOpModePrefTlv</a>	188
<a href="#">nas_CommInfo</a>	188
<a href="#">nas_CsgId</a>	190
<a href="#">nas_CSGID</a>	190
<a href="#">nas_currentPLMN</a>	191
<a href="#">nas_dataSrvCapabilities</a>	192
<a href="#">nas_DataStatusDetail</a>	193
<a href="#">nas_detailSvcInfo</a>	195
<a href="#">nas_DeviceConfigDetail</a>	196
<a href="#">nas_dirNum</a>	197
<a href="#">nas_DRCPParams</a>	197
<a href="#">nas_ecioListElement</a>	198
<a href="#">nas_ECIOThresh</a>	199
<a href="#">nas_EdrxCiotLteMode</a>	199
<a href="#">nas_EdrxCycleLength</a>	200
<a href="#">nas_EdrxEnableType</a>	200
<a href="#">nas_EdrxPagingTimeWindow</a>	201
<a href="#">nas_EdrxRatType</a>	201
<a href="#">nas_EmerModeTlv</a>	202
<a href="#">nas_errorRateListElement</a>	203
<a href="#">nas_ForbiddenNetworks3GPP</a>	204
<a href="#">nas_GERANInfo</a>	205
<a href="#">nas_geranInstArr</a>	206
<a href="#">nas_geranInstInfo</a>	207
<a href="#">nas_gsmCellInfo</a>	208
<a href="#">nas_GSMRSSIThresh</a>	209
<a href="#">nas_GSMSrvStatusInfo</a>	210
<a href="#">nas_GSMSysInfo</a>	211
<a href="#">nas_GWAcqOrderPrefTlv</a>	213
<a href="#">nas_HDRECIOThresh</a>	214
<a href="#">nas_HDRIOThresh</a>	214
<a href="#">nas_HDRPersonality_Ind_Data</a>	215
<a href="#">nas_HDRRSSIThresh</a>	216
<a href="#">nas_HDRSINRThresh</a>	216
<a href="#">nas_HDRSINRThreshold</a>	217
<a href="#">nas_HDRSysInfo</a>	218
<a href="#">nas_homeNwMNC3GppTlv</a>	220
<a href="#">nas_homeSIDNID</a>	220
<a href="#">nas_ImsVoiceSupportLteTlv</a>	221
<a href="#">nas_infoInterFreq</a>	221
<a href="#">nas_IOThresh</a>	223
<a href="#">nas_lteBandPrefExt</a>	223
<a href="#">nas_LTEBandPrefTlv</a>	224
<a href="#">nas_LteCiotOpModeTlv</a>	226
<a href="#">nas_lteEARFCN</a>	226
<a href="#">nas_LteEarfcnInfo</a>	227

<a href="#">nas_LteEmbmsCoverageTlv</a>	227
<a href="#">nas_LteEmbmsTraceIdTlv</a>	228
<a href="#">nas_LteGsmCellInfo</a>	229
<a href="#">nas_LTEInfo</a>	230
<a href="#">nas_LTEInfoInterfreq</a>	232
<a href="#">nas_LTEInfoIntrafreq</a>	232
<a href="#">nas_LTEInfoNeighboringGSM</a>	234
<a href="#">nas_LTEInfoNeighboringWCDMA</a>	235
<a href="#">nas_LteM1BandPrefTlv</a>	236
<a href="#">nas_LteNb1BandPrefTlv</a>	238
<a href="#">nas_LTEOperationalModeTlv</a>	240
<a href="#">nas_LTEOperationMode</a>	240
<a href="#">nas_LteOpMode</a>	241
<a href="#">nas_LteOpModeTlv</a>	241
<a href="#">nas_LtePCI</a>	242
<a href="#">nas_LteRegDomainTlv</a>	243
<a href="#">nas_LteRsrpInformation</a>	244
<a href="#">nas_LTERSRPThresh</a>	244
<a href="#">nas_LTERSQRThresh</a>	245
<a href="#">nas_LTERSSIThresh</a>	245
<a href="#">nas_LTESigRptCfg</a>	246
<a href="#">nas_LTESigRptConfig</a>	247
<a href="#">nas_LteSnrinformation</a>	248
<a href="#">nas_LTESNRThresh</a>	248
<a href="#">nas_LTESNRThreshold</a>	249
<a href="#">nas_LTESysInfo</a>	250
<a href="#">nas_LteVoiceDomainTlv</a>	252
<a href="#">nas_LteWcdmaCellInfo</a>	253
<a href="#">nas_Mdn</a>	254
<a href="#">nas_minBasedIMSI</a>	254
<a href="#">nas_MNCPCSDigitStatus</a>	255
<a href="#">nas_MNRInfo</a>	256
<a href="#">nas_ModePrefTlv</a>	256
<a href="#">nas_namName</a>	257
<a href="#">nas_netSelectionPref</a>	258
<a href="#">nas_NetSelPrefTlv</a>	258
<a href="#">nas_networkNameSrcTlv</a>	259
<a href="#">nas_NetworkStat1x</a>	260
<a href="#">nas_NetworkStatEVDO</a>	262
<a href="#">nas_nmrCellInfo</a>	263
<a href="#">nas_nr5gBandPref</a>	265
<a href="#">nas_NR5GCellStatusInfoTlv</a>	265
<a href="#">nas_NR5GCellStatusTlv</a>	266
<a href="#">nas_NR5GSerStatTlv</a>	267
<a href="#">nas_NR5GSrvStatusTlv</a>	268
<a href="#">nas_NR5GSysInfoTlv</a>	269
<a href="#">nas_NR5GSystemInfoTlv</a>	272
<a href="#">nas_NumScellsConfig</a>	276
<a href="#">nas_nwNameSrc3GppTlv</a>	276
<a href="#">nas_operatorNameString</a>	277
<a href="#">nas_OperatorPLMNData</a>	277
<a href="#">nas_operatorPLMNList</a>	278
<a href="#">nas_PhyCaAggDIBW</a>	279
<a href="#">nas_PhyCaAggPcellInfo</a>	279
<a href="#">nas_PhyCaAggScellDIBw</a>	280
<a href="#">nas_PhyCaAggScellIndex</a>	281
<a href="#">nas_PhyCaAggScellIndType</a>	281
<a href="#">nas_PhyCaAggScellInfo</a>	282

<a href="#">nas_PilotSetData</a>	284
<a href="#">nas_PilotSetParams</a>	285
<a href="#">nas_PlmnID</a>	285
<a href="#">nas_PLMNNetworkName</a>	286
<a href="#">nas_PLMNNetworkNameData</a>	286
<a href="#">nas_PRLPrefTlv</a>	288
<a href="#">nas_protocolSubtypeElement</a>	289
<a href="#">nas_qaQmi3Gpp2TimeZone</a>	290
<a href="#">nas_QmiNas3GppNetworkInfo</a>	291
<a href="#">nas_QmiNas3GppNetworkRAT</a>	292
<a href="#">nas_QmisNasPcsDigit</a>	293
<a href="#">nas_QmisNasSlqsNasPCICellInfo</a>	294
<a href="#">nas_QmisNasSlqsNasPCIInfo</a>	295
<a href="#">nas_RankIndicatorTlv</a>	296
<a href="#">nas_RatDisabledMaskTlv</a>	296
<a href="#">nas_RejectReasonTlv</a>	297
<a href="#">nas_RFBandInfoElements</a>	297
<a href="#">nas_RfBandInfoExtFormat</a>	298
<a href="#">nas_RfBandInfoExtFormatElements</a>	299
<a href="#">nas_RFBandInfoExtTlv</a>	300
<a href="#">nas_RfBandInfoList</a>	301
<a href="#">nas_RFBandInfoTlv</a>	301
<a href="#">nas_RfBandwidthInfo</a>	302
<a href="#">nas_RfBandwidthInfoElements</a>	303
<a href="#">nas_RFBandwidthInfoTlv</a>	304
<a href="#">nas_RfDedicatedBandInfo</a>	305
<a href="#">nas_RfDedicatedBandInfoElements</a>	306
<a href="#">nas_RFDedicatedBandInfoTlv</a>	307
<a href="#">nas_RFInfoTlv</a>	308
<a href="#">nas_roamIndList</a>	309
<a href="#">nas_RoamPrefTlv</a>	310
<a href="#">nas_RSRPThresh</a>	310
<a href="#">nas_rsrqInformation</a>	311
<a href="#">nas_RSRQThresh</a>	311
<a href="#">nas_RSSIThresh</a>	312
<a href="#">nas_rxInfo</a>	313
<a href="#">nas_RxSigInfo</a>	314
<a href="#">nas_rxSignalStrengthListElement</a>	315
<a href="#">nas_SccRxInfo</a>	316
<a href="#">nas_serviceProviderName</a>	317
<a href="#">nas_servSystem</a>	318
<a href="#">nas_sidNid</a>	319
<a href="#">nas_SignalStrengthTlv</a>	320
<a href="#">nas_SimRejInfoTlv</a>	320
<a href="#">nas_SLQSSignalStrengthsIndReq</a>	321
<a href="#">nas_SLQSSignalStrengthsInformation</a>	322
<a href="#">nas_SLQSSignalStrengthsTlv</a>	324
<a href="#">nas_SrvDomainPrefTlv</a>	324
<a href="#">nas_SrvRegRestrictionTlv</a>	325
<a href="#">nas_SrvRegRestrictTlv</a>	326
<a href="#">nas_SrvStatusInfo</a>	326
<a href="#">nas_sysInfoCommon</a>	327
<a href="#">nas_TDSCDMABandPrefTlv</a>	329
<a href="#">nas_TDSCDMAECIOThresh</a>	330
<a href="#">nas_TDSCDMARSCPTthresh</a>	330
<a href="#">nas_TDSCDMARSSIThresh</a>	331
<a href="#">nas_TDSCDMASINRCONFThresh</a>	331
<a href="#">nas_TDSCDMASINRThresh</a>	332



nas_timeInfo	332
nas_trueIMSI	334
nas_txInfo	335
nas_UMTSExtInfo	335
nas_UMTSInfo	337
nas_umtsInstArr	339
nas_UMTSinstInfo	340
nas_umtsLTENbrCell	341
nas_UniversalTime	342
nas_UsageSettingTlv	343
nas_VoiceDomainPrefTlv	344
nas_wcdmaCellInfo	344
nas_WCDMACellInfoExt	345
nas_WCDMAECIOThresh	346
nas_WCDMAInfoLTENeighborCell	347
nas_WCDMARSSIThresh	347
nas_WCDMASysInfo	348
nas_wcdmaUARFCN	351
NASAcqOrderPrefTlv	352
NASBandPreferenceTlv	352
NASCIotAcqOrderPrefTlv	353
NASCIotLteOpModePrefTlv	353
NASEmergencyModeTlv	354
NasGetLTECphyCaInfo	354
NASGWAcqOrderPrefTlv	355
NASLTEBandPreferenceTlv	356
NASLteM1BandPrefTlv	356
NASLteNasReleaseInfoTlv	357
NASLteNB1BandPrefTlv	357
NASModePreferenceTlv	358
NASNetSelPreferenceTlv	358
NASNr5gBandPrefTlv	359
NASOTAMessageTlv	359
NASPhyCaAggPcellInfo	360
NASPhyCaAggScellArray	361
NASPhyCaAggScellIDBw	362
NASPhyCaAggScellIndex	363
NASPhyCaAggScellIndType	363
NASPhyCaAggScellInfo	364
NASPRLPreferenceTlv	365
NASQmiCbkNasSwiOTAMessageInd	366
NASQmiCbkNasSystemSelPrefInd	366
NASRatDisabledMaskTlv	369
NASRoamPreferenceTlv	369
NASServDomainPrefTlv	370
NASServingSystemInfo	370
NASTimeInfoTlv	371
newMTMessageTlv	372
pack_audio_SLQSGetAudioPathConfig_t	372
pack_audio_SLQSGetAudioProfile_t	373
pack_audio_SLQSGetAudioVoTLBConfig_t	374
pack_audio_SLQSSetAudioPathConfig_t	374
pack_audio_SLQSSetAudioProfile_t	376
pack_audio_SLQSSetAudioVoTLBConfig_t	378
pack_cat_CATSendEnvelopeCommand_t	378
pack_cat_CATSendTerminalResponse_t	379
pack_cat_SetCATEventCallback_t	380
pack_dms_ActivateAutomatic_t	381

pack_dms_GetCustFeaturesV2_t	381
pack_dms_ResetToFactoryDefaults_t	382
pack_dms_SetActivationStatusCallback_t	382
pack_dms_SetCrashAction_t	382
pack_dms_SetCustFeature_t	383
pack_dms_SetCustFeaturesV2_t	385
pack_dms_SetEventReport_t	386
pack_dms_SetIndicationRegister_t	386
pack_dms_SetPower_t	387
pack_dms_SetUSBComp_t	388
pack_dms_SLQSDmsSwiIndicationRegister_t	389
pack_dms_SLQSSetPowerSaveModeConfig_t	389
pack_dms_SLQSSwiGetCrashInfo_t	390
pack_dms_SLQSSwiSetDyingGaspCfg_t	390
pack_dms_SLQSSwiSetHostDevInfo_t	391
pack_dms_SLQSSwiSetOSInfo_t	392
pack_dms_SwiSetEventReport_t	392
pack_dms_SwiUimSelect_t	393
pack_dms_UIMChangePIN_t	394
pack_dms_UIMGetControlKeyStatus_t	394
pack_dms_UIMGetICCID_t	395
pack_dms_UIMSetControlKeyProtection_t	395
pack_dms_UIMSetPINProtection_t	396
pack_dms_UIMUnblockControlKey_t	397
pack_dms_UIMUnblockPIN_t	397
pack_dms_UIMVerifyPIN_t	398
pack_fms_GetImagesPreference_t	398
pack_fms_GetStoredImages_t	399
pack_fms_SetImagesPreference_t	399
pack_ims_SLQSImsConfigIndicationRegister_t	400
pack_ims_SLQSSetIMSSMSCConfig_t	401
pack_ims_SLQSSetIMSUserConfig_t	402
pack_ims_SLQSSetIMSVoIPConfig_t	403
pack_ims_SLQSSetRegMgrConfig_t	405
pack_ims_SLQSSetSIPConfig_t	406
pack_imsa_SLQSRegisterIMSAIndication_t	407
pack_loc_Delete_Assist_Data_t	408
pack_loc_EventRegister_t	409
pack_loc_SetExtPowerState_t	411
pack_loc_SetOperationMode_t	411
pack_loc_SLQSLOCGetBestAvailPos_t	412
pack_loc_SLQSLOCGetServer_t	413
pack_loc_SLQSLOCInjectPosition_t	413
pack_loc_SLQSLOCInjectSensorData_t	418
pack_loc_SLQSLOCInjectUTCTime_t	420
pack_loc_SLQSLOCSetCradleMountConfig_t	420
pack_loc_SLQSLOCSetServer_t	421
pack_loc_Start_t	422
pack_loc_Stop_t	424
pack_nas_InitiateDomainAttach_t	424
pack_nas_PerformNetworkScanPCI_t	425
pack_nas_SetACCOLC_t	428
pack_nas_SetCDMANetworkParameters_t	428
pack_nas_SetNetworkPreference_t	430
pack_nas_SLQSConfigSigInfo_t	431
pack_nas_SLQSGetPLMNName_t	432
pack_nas_SLQSInitiateNetworkRegistration_t	433
pack_nas_SLQSNasConfigSigInfo2_t	434

pack_nas_SLQSNasGet3GPP2Subscription_t	439
pack_nas_SLQSNASGeteDRXParamsExt_t	439
pack_nas_SLQSNasGetTxRxInfo_t	440
pack_nas_SLQSNasIndicationRegisterExt_t	440
pack_nas_SLQSNasIndicationRegisterV2_t	443
pack_nas_SLQSNASSeteDRXParams_t	451
pack_nas_SLQSNasSwiIndicationRegister_t	453
pack_nas_SLQSNASSwiSetChannelLock_t	454
pack_nas_SLQSSetBandPreference_t	455
pack_nas_SLQSSetSignalStrengthsCallback_t	457
pack_nas_SLQSSetSysSelectionPref_t	457
pack_nas_SLQSSetSysSelectionPrefExt_t	463
pack_nas_SLQSSwiPSDetach_t	473
pack_pds_PDSInjectTimeReference_t	473
pack_pds_ResetPDSDData_t	474
pack_pds_SetEventReportCallback_t	475
pack_pds_SetPDSDefaults_t	475
pack_pds_SetPDSState_t	476
pack_pds_SetPortAutomaticTracking_t	476
pack_pds_SetServiceAutomaticTracking_t	477
pack_pds_SetXTRAAutomaticDownload_t	477
pack_pds_SetXTRANetwork_t	478
pack_pds_SLQSGetAGPSConfig_t	478
pack_pds_SLQSPDSInjectAbsoluteTimeReference_t	479
pack_pds_SLQSPDSInjectPositionData_t	480
pack_pds_SLQSSetAGPSConfig_t	482
pack_pds_SLQSSetPositionMethodState_t	482
pack_pds_StartPDSTrackingSessionExt_t	483
pack_qmi_t	484
pack_qos_BindDataPort_t	485
pack_qos_SLQSQosSwiReadApnExtraParams_t	486
pack_qos_SLQSQosSwiReadDataStats_t	486
pack_qos_SLQSSetQosEventCallback_t	486
pack_rms_SetSMSWake_t	487
pack_sar_SLQSSetRfSarState_t	487
pack_sms_SaveSMS_t	488
pack_sms_SendSMS_t	489
pack_sms_SetNewSMSCallback_t	490
pack_sms_SetSMSCAddress_t	490
pack_sms_SLQSDeleteSMS_t	490
pack_sms_SLQSGetSMS_t	491
pack_sms_SLQSGetSmsBroadcastConfig_t	492
pack_sms_SLQSGetSMSList_t	492
pack_sms_SLQSModifySMSStatus_t	493
pack_sms_SLQSSendAsyncSMS_t	494
pack_sms_SLQSSetIndicationRegister_t	494
pack_sms_SLQSSetSmsBroadcastActivation_t	495
pack_sms_SLQSSetSmsBroadcastConfig_t	495
pack_sms_SLQSSetSmsStorage_t	496
pack_sms_SLQSSmsGetMaxStorageSize_t	497
pack_sms_SLQSSmsSetRoutes_t	497
pack_swiaudio_SLQSGetM2MAudioProfile_t	497
pack_swiaudio_SLQSGetM2MAudioVolume_t	498
pack_swiaudio_SLQSGetM2MAVMute_t	498
pack_swiaudio_SLQSGetM2MSpkrGain_t	499
pack_swiaudio_SLQSSetM2MAudioAVCFG_t	499
pack_swiaudio_SLQSSetM2MAudioLPBK_t	500
pack_swiaudio_SLQSSetM2MAudioProfile_t	500

pack_swiaudio_SLQSSetM2MAudioVolume_t	502
pack_swiaudio_SLQSSetM2MAVMute_t	502
pack_swiaudio_SLQSSetM2MSpkrGain_t	503
pack_swiaavms_SLQSAVMSSendSelection_t	504
pack_swiaavms_SLQSAVMSSetSettings_t	504
pack_swiaavms_SLQSAVMSSetSettings_v2_t	506
pack_swiaavms_SLQSAVMSStartSession_t	508
pack_swiaavms_SLQSAVMSStopSession_t	508
pack_swidms_SLQSSwiDmsSetHWWatchdog_t	509
pack_swidms_SLQSSwiDmsSetMTU_t	509
pack_swidms_SLQSSwiDmsSetUsbComp_t	510
pack_swidms_SLQSSwiDmsSetUsbNetNum_t	511
pack_swiloc_SwiLocSetAutoStart_t	512
pack_swioma_SLQSOMADMCancelSession_t	513
pack_swioma_SLQSOMADMCancelSessionExt_t	514
pack_swioma_SLQSOMADMGetSessionInfo_t	514
pack_swioma_SLQSOMADMSendSelection_t	515
pack_swioma_SLQSOMADMSendSelectionExt_t	515
pack_swioma_SLQSOMADMSetSettings_t	516
pack_swioma_SLQSOMADMSetSettingsExt_t	517
pack_swioma_SLQSOMADMStartSession_t	518
pack_swioma_SLQSOMADMStartSessionExt_t	518
pack_tmd_SLQSTmdDeRegNotMitigationLvl_t	519
pack_tmd_SLQSTmdGetMitigationLvl_t	519
pack_tmd_SLQSTmdRegNotMitigationLvl_t	520
pack_uim_ChangePin_t	520
pack_uim_ReadTransparent_t	521
pack_uim_SetPinProtection_t	523
pack_uim_SLQSUIMAuthenticate_t	524
pack_uim_SLQSUIMDepersonalization_t	524
pack_uim_SLQSUIMEventRegister_t	525
pack_uim_SLQSUIMGetConfiguration_t	525
pack_uim_SLQSUIMGetFileAttributes_t	526
pack_uim_SLQSUIMGetServiceStatus_t	526
pack_uim_SLQSUIMPowerDown_t	527
pack_uim_SLQSUIMPowerUp_t	527
pack_uim_SLQSUIMReadRecord_t	528
pack_uim_SLQSUIMRefreshComplete_t	529
pack_uim_SLQSUIMRefreshGetLastEvent_t	529
pack_uim_SLQSUIMRefreshOK_t	530
pack_uim_SLQSUIMRefreshRegister_t	530
pack_uim_SLQSUIMSetServiceStatus_t	531
pack_uim_SLQSUIMSwitchSlot_t	531
pack_uim_SLQSUIMWriteRecord_t	532
pack_uim_SLQSUIMWriteTransparent_t	533
pack_uim_UnblockPin_t	534
pack_uim_VerifyPin_t	535
pack_voice_AnswerUSSD_t	536
pack_voice_OriginateUSSD_t	536
pack_voice_SLQSOriginateUSSD_t	537
pack_voice_SLQSVoiceALSSelectLine_t	537
pack_voice_SLQSVoiceALSSetLineSwitching_t	538
pack_voice_SLQSVoiceAnswerCall_t	538
pack_voice_SLQSVoiceBindSubscription_t	539
pack_voice_SLQSVoiceBurstDTMF_t	539
pack_voice_SLQSVoiceDialCall_t	540
pack_voice_SLQSVoiceEndCall_t	541
pack_voice_SLQSVoiceGetCallBarring_t	542

pack_voice_SLQSVoiceGetCallForwardingStatus_t	543
pack_voice_SLQSVoiceGetCallInfo_t	543
pack_voice_SLQSVoiceGetCallWaiting_t	544
pack_voice_SLQSVoiceGetConfig_t	544
pack_voice_SLQSVoiceIndicationRegister_t	546
pack_voice_SLQSVoiceManageCalls_t	547
pack_voice_SLQSVoiceOrigUSSDNoWait_t	548
pack_voice_SLQSVoiceSendFlash_t	548
pack_voice_SLQSVoiceSetCallBarringPassword_t	549
pack_voice_SLQSVoiceSetConfig_t	550
pack_voice_SLQSVoiceSetPreferredPrivacy_t	551
pack_voice_SLQSVoiceSetSUPSService_t	551
pack_voice_SLQSVoiceStartContDTMF_t	554
pack_voice_SLQSVoiceStopContDTMF_t	555
pack_wds_DHCPv4ClientLeaseChange_t	555
pack_wds_GetDefaultProfile_t	555
pack_wds_GetDefaultProfileNum_t	556
pack_wds_GetDefaultProfileV2_t	556
pack_wds_GetDormancyState_t	557
pack_wds_GetLastMobileIPError_t	557
pack_wds_GetMobileIP_t	557
pack_wds_GetMobileIPProfile_t	558
pack_wds_GetPacketStatistics_t	558
pack_wds_GetPacketStatus_t	558
pack_wds_GetSessionDuration_t	559
pack_wds_RMSetTransferStatistics_t	559
pack_wds_SetAutoconnect_t	560
pack_wds_SetDefaultProfile_t	560
pack_wds_SetDefaultProfileNum_t	562
pack_wds_SetMobileIP_t	563
pack_wds_SetMobileIPParameters_t	563
pack_wds_SetMobileIPProfile_t	564
pack_wds_SLQSCreateProfile_t	566
pack_wds_SLQSDeleteProfile_t	567
pack_wds_SLQSGetCurrDataSystemStat_t	568
pack_wds_SLQSGetDataBearerTechnology_t	568
pack_wds_SLQSGetDUNCallInfo_t	568
pack_wds_SLQSGetProfileSettings_t	570
pack_wds_SLQSGetRuntimeSettings_t	570
pack_wds_SLQSModifyProfile_t	571
pack_wds_SLQSSet3GPPConfigItem_t	572
pack_wds_SLQSSetIPFamilyPreference_t	574
pack_wds_SLQSSetWdsEventCallback_t	574
pack_wds_SLQSSetDHCPv4ClientConfig_t	575
pack_wds_SLQSSetDHCPv4ClientConfig_t	575
pack_wds_SLQSSetLoopback_t	576
pack_wds_SLQSStartDataSession_t	576
pack_wds_SLQSStopDataSession_t	578
pack_wds_SLQSSwiProfileChangeCallback_t	578
pack_wds_SLQSWdsSetEventReport_t	578
pack_wds_SLQSWdsSwiPDPRuntimeSettings_t	580
PackCreateProfileOut	581
packgetDyingGaspCfg	581
packgetDyingGaspStatistics	582
PackSwiAvmsSetSettingsAPNInfo	582
PackSwiAvmsSetSettingsConnectionRetryTimers	583
PackSwiAvmsSetSettingsPeriodInfo	584
PackSwiAVMSSettingsAPNInfo	584

PackSwiAVMSSettingsConnectionRetryTimers	585
PackSwiAVMSSettingsPeriodsInfo	586
qmiSmsMessageList	586
qmiWSDDataBearerTechnology	587
qmTlvResult	587
qos_BindDataPortMuxID_t	588
qos_BindDataPortPeripheralEndPointID_t	588
qos_BindDataPortSIODDataPort_t	589
RFBandInfoElements	589
rmTrasnferStaticsReq	590
sensorData_t	590
slot_t	592
slotInf	593
slots_t	594
sms_BroadcastConfig	595
sms_CDMABroadcastConfig	595
sms_getIndicationReg	596
sms_getMsgWaitingInfo	597
sms_getTransLayerInfo	597
sms_getTransNWRegInfo	598
sms_maxStorageSizeReq	599
sms_maxStorageSizeResp	599
sms_messageWaitingInfoContent	600
sms_msgProtocolResp	601
sms_qaQmi3GPP2BroadcastCfgInfo	601
sms_qaQmi3GPPBroadcastCfgInfo	602
sms_routeEntry	602
sms_sendAsynCsmsParams	603
sms_setIndicationReg	606
sms_setRoutesReq	606
sms_transLayerInfo	607
sMSCAddressInfo	608
sMSCAddressTlv	608
sMSEtsMessageInfo	609
sMSEtsMessageTlv	609
sMSEtsPlmnInfo	610
sMSMessageModelInfo	610
sMSMTMessageInfo	611
sMSOnIMSInfo	611
sMSOnIMSTlv	612
sMSTransferRouteMTMessageInfo	612
swi_uint256_t	613
swiaudio_PCMparams	613
swidms_ehrpdMTUSizeTlv	614
swidms_hrpdmTUSizeTlv	614
swidms_ifaceCfgTlv	615
swidms_mtuSize3gppTlv	616
swidms_supportedIntBitmaskTlv	617
swidms_SwiDmsGetHWWatchdog	617
swidms_usbMTUSizeTlv	618
tdscdmaSigInfoExt	619
tempData_t	619
tmd_mitigationDevList	620
transferRouteMessageTlv	621
uim_additionalReadResult	621
uim_appStatus	622
uim_authenticateResult	625
uim_authenticationData	626

uim_cardResult	627
uim_cardResultInfo	627
uim_cardStatus	628
uim_changeUIMPIN	629
uim_depersonalizationInformation	630
uim_encryptedPIN1	631
uim_fileAttributes	631
uim_fileInfo	635
uim_GetSlotsInfoTlv	635
uim_GetSlotsStatusTlv	636
uim_hotSwapStatus	637
uim_indToken	637
uim_personalizationStatus	638
uim_physlotInfo	639
uim_physlotsInfo	639
uim_readRecordInfo	640
uim_readResult	640
uim_readResultInfo	641
uim_readTransparentInfo	642
uim_refreshevent	642
uim_registerRefresh	644
uim_remainingRetries	645
uim_sessionInformation	645
uim_setPINProtection	646
uim_simBusyStatus	647
uim_slotInfo	648
uim_UIMGetFDNStatus	649
uim_UIMGetHiddenKeyStatus	650
uim_UIMGetIndex	650
uim_UIMSessionInformation	651
uim_unblockUIMPIN	652
uim_validCardStatus	653
uim_verifyUIMPIN	653
uim_writeRecordInfo	654
uim_writeTransparentInfo	654
unpack_audio_SLQSGetAudioPathConfig_t	655
unpack_audio_SLQSGetAudioProfile_t	657
unpack_audio_SLQSGetAudioVolTLBConfig_t	658
unpack_audio_SLQSSetAudioVolTLBConfig_t	659
unpack_cat_SetCatEventCallback_ind_t	659
unpack_cat_SetCATEventCallback_t	660
unpack_dms_GetActivationState_t	661
unpack_dms_GetBandCapability_t	662
unpack_dms_GetCrashAction_t	664
unpack_dms_GetCustFeature_t	665
unpack_dms_GetCustFeaturesV2_t	667
unpack_dms_GetDeviceCap_t	668
unpack_dms_GetDeviceCapabilities_t	669
unpack_dms_GetDeviceCapabilitiesV2_t	671
unpack_dms_GetDeviceHardwareRev_t	673
unpack_dms_GetDeviceMfr_t	674
unpack_dms_GetDeviceSerialNumbers_t	675
unpack_dms_GetFirmwareInfo_t	677
unpack_dms_GetFirmwareRevision_t	679
unpack_dms_GetFirmwareRevisions_t	680
unpack_dms_GetFSN_t	681
unpack_dms_GetHardwareRevision_t	682
unpack_dms_GetIMSI_t	682

<a href="#">unpack_dms_GetManufacturer_t</a>	683
<a href="#">unpack_dms_GetModelID_t</a>	683
<a href="#">unpack_dms_GetNetworkTime_t</a>	684
<a href="#">unpack_dms_GetNetworkTimeV2_t</a>	685
<a href="#">unpack_dms_GetOfflineReason_t</a>	686
<a href="#">unpack_dms_GetPower_t</a>	687
<a href="#">unpack_dms_GetPRLVersion_t</a>	688
<a href="#">unpack_dms_GetSerialNumbers_t</a>	689
<a href="#">unpack_dms_GetUSBComp_t</a>	690
<a href="#">unpack_dms_GetVoiceNumber_t</a>	692
<a href="#">unpack_dms_PSMCfgChange_ind_t</a>	694
<a href="#">unpack_dms_ResetToFactoryDefaults_t</a>	694
<a href="#">unpack_dms_SetActivationStatusCallback_t</a>	695
<a href="#">unpack_dms_SetCrashAction_t</a>	695
<a href="#">unpack_dms_SetCustFeature_t</a>	696
<a href="#">unpack_dms_SetCustFeaturesV2_t</a>	696
<a href="#">unpack_dms_SetEventReport_ind_t</a>	697
<a href="#">unpack_dms_SetEventReport_t</a>	698
<a href="#">unpack_dms_SetFirmwarePreference_t</a>	698
<a href="#">unpack_dms_SetIndicationRegister_t</a>	699
<a href="#">unpack_dms_SetPower_t</a>	699
<a href="#">unpack_dms_SetUSBComp_t</a>	700
<a href="#">unpack_dms_SLQSDmsSwiGetPCInfo_t</a>	700
<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t</a>	703
<a href="#">unpack_dms_SLQSDmsSwiGetResetInfo_t</a>	704
<a href="#">unpack_dms_SLQSDmsSwiGetUimSelection_t</a>	705
<a href="#">unpack_dms_SLQSDmsSwiIndicationRegister_t</a>	706
<a href="#">unpack_dms_SLQSGetBandCapability_t</a>	707
<a href="#">unpack_dms_SLQSGetBandCapabilityExt_t</a>	711
<a href="#">unpack_dms_SLQSGetERIFile_t</a>	715
<a href="#">unpack_dms_SLQSGetPowerSaveModeConfig_t</a>	716
<a href="#">unpack_dms_SLQSSetPowerSaveModeConfig_t</a>	717
<a href="#">unpack_dms_SLQSSwiClearDyingGaspStatistics_t</a>	718
<a href="#">unpack_dms_SLQSSwiGetCrashInfo_t</a>	718
<a href="#">unpack_dms_SLQSSwiGetDyingGaspCfg_t</a>	719
<a href="#">unpack_dms_SLQSSwiGetDyingGaspStatistics_t</a>	720
<a href="#">unpack_dms_SLQSSwiGetFirmwareCurr_t</a>	720
<a href="#">unpack_dms_SLQSSwiGetFwUpdateStatus_t</a>	721
<a href="#">unpack_dms_SLQSSwiGetHostDevInfo_t</a>	723
<a href="#">unpack_dms_SLQSSwiGetOSInfo_t</a>	724
<a href="#">unpack_dms_SLQSSwiGetSerialNoExt_t</a>	725
<a href="#">unpack_dms_SLQSSwiSetDyingGaspCfg_t</a>	726
<a href="#">unpack_dms_SLQSSwiSetHostDevInfo_t</a>	726
<a href="#">unpack_dms_SLQSSwiSetOSInfo_t</a>	727
<a href="#">unpack_dms_SLQSUIMGetState_t</a>	727
<a href="#">unpack_dms_SwiEventReportCallBack_ind_t</a>	728
<a href="#">unpack_dms_SwiSetEventReport_t</a>	729
<a href="#">unpack_dms_SwiUimSelect_t</a>	729
<a href="#">unpack_dms_UIMGetControlKeyStatus_t</a>	730
<a href="#">unpack_dms_UIMGetICCID_t</a>	731
<a href="#">unpack_dms_UIMGetPINStatus_t</a>	732
<a href="#">unpack_dms_UIMSetControlKeyProtection_t</a>	733
<a href="#">unpack_dms_UIMSetPINProtection_t</a>	734
<a href="#">unpack_dms_UIMUnblockControlKey_t</a>	735
<a href="#">unpack_fms_GetImagesPreference_t</a>	735
<a href="#">unpack_fms_GetStoredImages_t</a>	736
<a href="#">unpack_fms_SetImagesPreference_t</a>	737
<a href="#">unpack_ims_SLQSGetIMSSMSConfig_t</a>	737



unpack_ims_SLQSGetIMSUserConfig_t	738
unpack_ims_SLQSGetIMSVoIPConfig_t	739
unpack_ims_SLQSGetRegMgrConfig_t	742
unpack_ims_SLQSGetSIPConfig_t	743
unpack_ims_SLQSRegMgrCfgCallBack_ind_t	744
unpack_ims_SLQSSetIMSSMSConfig_t	745
unpack_ims_SLQSSetIMSUserConfig_t	746
unpack_ims_SLQSSetIMSVoIPConfig_t	746
unpack_ims_SLQSSetRegMgrConfig_t	747
unpack_ims_SLQSSetSIPConfig_t	747
unpack_ims_SLQSSIPCfgCallBack_ind_t	748
unpack_ims_SLQSSMSCfgCallBack_ind_t	749
unpack_ims_SLQSUserCfgCallBack_ind_t	750
unpack_ims_SLQSVoIPCfgCallBack_ind_t	751
unpack_imsa_SLQSGetIMSARegStatus_t	753
unpack_imsa_SLQSGetIMSAServiceStatus_t	754
unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t	756
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t	757
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t	758
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t	759
unpack_loc_BestAvailPos_Ind_t	760
unpack_loc_CradleMountCallback_Ind_t	767
unpack_loc_Delete_Assist_Data_t	767
unpack_loc_DeleteAssistData_Ind_t	768
unpack_loc_EngineState_Ind_t	769
unpack_loc_EventNMEA_Ind_t	770
unpack_loc_EventRegister_t	771
unpack_loc_EventTimeSyncCallback_Ind_t	771
unpack_loc_FixCriteria_Ind_t	771
unpack_loc_GetOpMode_Ind_t	773
unpack_loc_GetServer_Ind_t	774
unpack_loc_GnssSvInfo_Ind_t	775
unpack_loc_InjectPositionCallback_Ind_t	776
unpack_loc_InjectSensorDataCallback_Ind_t	777
unpack_loc_InjectTimeSyncDataCallback_Ind_t	779
unpack_loc_InjectUTCTimeCallback_Ind_t	779
unpack_loc_PositionRpt_Ind_t	780
unpack_loc_SensorStreamingCallback_Ind_t	787
unpack_loc_SetExtPowerConfig_Ind_t	788
unpack_loc_SetExtPowerState_t	788
unpack_loc_SetOperationMode_Ind_t	789
unpack_loc_SetOperationMode_t	790
unpack_loc_SetServer_Ind_t	790
unpack_loc_SLQSLOCGetBestAvailPos_t	791
unpack_loc_SLQSLOCGetOpMode_t	792
unpack_loc_Start_t	792
unpack_loc_Stop_t	793
unpack_nas_GetACCOLC_t	793
unpack_nas_GetANAAAAuthenticationStatus_t	794
unpack_nas_GetCDMANetworkParameters_t	794
unpack_nas_GetHomeNetwork3GPP2_t	797
unpack_nas_GetHomeNetwork_t	799
unpack_nas_GetNetworkPreference_t	800
unpack_nas_GetRFInfo_t	801
unpack_nas_GetServingNetwork_t	802
unpack_nas_GetServingNetworkCapabilities_t	805
unpack_nas_GetSignalStrengths_t	806
unpack_nas_PerformNetworkScan_t	807

<a href="#">unpack_nas_SetDataCapabilitiesCallback_ind_t</a>	808
<a href="#">unpack_nas_SetEventReportInd_t</a>	809
<a href="#">unpack_nas_SetNasLTECphyCaIndCallback_ind_t</a>	810
<a href="#">unpack_nas_SetNetworkPreference_t</a>	811
<a href="#">unpack_nas_SetRoamingIndicatorCallback_ind_t</a>	812
<a href="#">unpack_nas_SetServingSystemCallback_ind_t</a>	812
<a href="#">unpack_nas_SLQSGetErrorRate_t</a>	813
<a href="#">unpack_nas_SLQSGetHomeNetwork_t</a>	814
<a href="#">unpack_nas_SlqsGetLTECphyCAInfo_t</a>	816
<a href="#">unpack_nas_SLQSGetNetworkTime_t</a>	816
<a href="#">unpack_nas_SLQSGetOperatorNameData_t</a>	817
<a href="#">unpack_nas_SLQSGetPLMNName_t</a>	818
<a href="#">unpack_nas_SLQSGetServingSystem_t</a>	822
<a href="#">unpack_nas_SLQSGetServingSystemV2_t</a>	826
<a href="#">unpack_nas_SLQSGetSignalStrength_t</a>	831
<a href="#">unpack_nas_SLQSGetSysInfo_t</a>	834
<a href="#">unpack_nas_SLQSGetSysInfoV2_t</a>	837
<a href="#">unpack_nas_SLQSGetSysSelectionPref_t</a>	842
<a href="#">unpack_nas_SLQSGetSysSelectionPrefExt_t</a>	847
<a href="#">unpack_nas_SLQSGetSysSelectionPrefExtV2_t</a>	850
<a href="#">unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t</a>	854
<a href="#">unpack_nas_SLQSNasGet3GPP2Subscription_t</a>	855
<a href="#">unpack_nas_SLQSNasGetCellLocationInfo_t</a>	856
<a href="#">unpack_nas_SLQSNasGetCellLocationInfoV2_t</a>	858
<a href="#">unpack_nas_SLQSNASGeteDRXParams_t</a>	860
<a href="#">unpack_nas_SLQSNASGeteDRXParamsExt_t</a>	861
<a href="#">unpack_nas_SLQSNASGetForbiddenNetworks_t</a>	862
<a href="#">unpack_nas_SLQSNasGetHDRColorCode_t</a>	863
<a href="#">unpack_nas_SLQSNasGetRFInfo_t</a>	863
<a href="#">unpack_nas_SLQSNasGetSigInfo_t</a>	864
<a href="#">unpack_nas_SLQSNasGetTxRxInfo_t</a>	865
<a href="#">unpack_nas_SLQSNasNetworkRejectCallback_Ind_t</a>	866
<a href="#">unpack_nas_SLQSNasNetworkTimeCallBack_ind_t</a>	869
<a href="#">unpack_nas_SLQSNasRFBandInfoCallback_Ind_t</a>	870
<a href="#">unpack_nas_SLQSNasSigInfoCallback_ind_t</a>	871
<a href="#">unpack_nas_SLQSNASSwiGetChannelLock_t</a>	873
<a href="#">unpack_nas_SLQSNasSwiModemStatus_t</a>	874
<a href="#">unpack_nas_SLQSNasSwiOTAMessageCallback_ind_t</a>	874
<a href="#">unpack_nas_SLQSNasTimerCallback_ind_t</a>	875
<a href="#">unpack_nas_SLQSPerformNetworkScanV2_t</a>	876
<a href="#">unpack_nas_SLQSSetSysSelectionPrefCallBack_ind_t</a>	878
<a href="#">unpack_nas_SLQSSwiGetHDRPersonality_t</a>	878
<a href="#">unpack_nas_SLQSSwiGetHDRProtSubtype_t</a>	879
<a href="#">unpack_nas_SLQSSwiGetHRPDStats_t</a>	880
<a href="#">unpack_nas_SLQSSwiGetLteCQI_t</a>	881
<a href="#">unpack_nas_SLQSSwiGetLteSccRxInfo_t</a>	882
<a href="#">unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t</a>	882
<a href="#">unpack_nas_SLQSSwiNetworkDebug_t</a>	883
<a href="#">unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t</a>	884
<a href="#">unpack_nas_SLQSSysInfoCallback_ind_t</a>	884
<a href="#">unpack_omaDmConfigTlv_t</a>	888
<a href="#">unpack_omaDmFotaTlv_t</a>	889
<a href="#">unpack_omaDmNotificationsTlv_t</a>	891
<a href="#">unpack_pds_GetPDSDefaults_t</a>	892
<a href="#">unpack_pds_GetPDSSState_t</a>	893
<a href="#">unpack_pds_GetPortAutomaticTracking_t</a>	894
<a href="#">unpack_pds_GetServiceAutomaticTracking_t</a>	895
<a href="#">unpack_pds_GetXTRAAutomaticDownload_t</a>	895

unpack_pds_GetXTRANetwork_t	896
unpack_pds_GetXTRAVality_t	896
unpack_pds_SetEventReport_Ind_t	897
unpack_pds_SetPdsState_Ind_t	899
unpack_pds_SLQSGetAGPSConfig_t	900
unpack_pds_SLQSGetGPSSStateInfo_t	901
unpack_qmi_t	906
unpack_qos_BindDataPort_t	906
unpack_qos_dataRate_t	907
unpack_qos_IPv4Addr_t	907
unpack_qos_IPv6Addr_t	907
unpack_qos_IPv6TrafCls_t	908
unpack_qos_pktErrRate_t	908
unpack_qos_Port_t	909
unpack_qos_QosFlowInfo_t	909
unpack_qos_QosFlowInfoState_t	911
unpack_qos_SLQSQosGetNetworkStatus_t	911
unpack_qos_SLQSQosSwiReadApnExtraParams_t	912
unpack_qos_SLQSQosSwiReadDataStats_t	914
unpack_qos_SLQSSetQosEventCallback_ind_t	915
unpack_qos_SLQSSetQosEventCallback_t	916
unpack_qos_SLQSSetQosNWStatusCallback_ind_t	916
unpack_qos_SLQSSetQosPriEventCallback_ind_t	917
unpack_qos_SLQSSetQosStatusCallback_ind_t	917
unpack_qos_swiQosFilter_t	919
unpack_qos_swiQosFlow_t	922
unpack_qos_tokenBucket_t	926
unpack_qos_Tos_t	927
unpack_QosFlowStat_t	927
unpack_result_t	928
unpack_rms_GetSMSWake_t	929
unpack_rms_SetSMSWake_t	929
unpack_RMTransferStatistics_ind_t	930
unpack_sar_SLQSGetRfSarState_t	931
unpack_sms_GetSMSCAddress_t	931
unpack_sms_SaveSMS_t	932
unpack_sms_SendSMS_t	933
unpack_sms_SetNewSMSCallback_ind_t	934
unpack_sms_SetNewSMSCallback_t	935
unpack_sms_SetSMSCAddress_t	935
unpack_sms_SLQSDeleteSMS_t	936
unpack_sms_SLQSGetIndicationRegister_t	936
unpack_sms_SLQSGetMessageWaiting_t	937
unpack_sms_SLQSGetSMS_t	937
unpack_sms_SLQSGetSmsBroadcastConfig_t	938
unpack_sms_SLQSGetSMSList_t	939
unpack_sms_SLQSGetTransLayerInfo_t	940
unpack_sms_SLQSGetTransNWRegInfo_t	940
unpack_sms_SLQSMModifySMSStatus_t	941
unpack_sms_SLQSNWRegInfoCallback_ind_t	941
unpack_sms_SLQSSendAsyncSMS_t	942
unpack_sms_SLQSSetIndicationRegister_t	942
unpack_sms_SLQSSetSmsBroadcastActivation_t	943
unpack_sms_SLQSSetSmsBroadcastConfig_t	943
unpack_sms_SLQSSetSmsStorage_t	944
unpack_sms_SLQSSmsGetMaxStorageSize_t	944
unpack_sms_SLQSSmsGetMessageProtocol_t	945
unpack_sms_SLQSSmsSetRoutes_t	945

unpack_sms_SLQSSwiGetSMSStorage_t	946
unpack_sms_SLQSTransLayerInfoCallback_ind_t	946
unpack_sms_SLQSWmsAsyncRawSendCallBack_ind_t	947
unpack_sms_SLQSWmsMemoryFullCallBack_ind_t	949
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t	950
unpack_swiaudio_SLQSGetM2MAudioProfile_t	950
unpack_swiaudio_SLQSGetM2MAudioVolume_t	952
unpack_swiaudio_SLQSGetM2MAVMute_t	952
unpack_swiaudio_SLQSGetM2MSpkrGain_t	953
unpack_swiaavms_SLQSAVMSEventReportInd_t	954
unpack_swiaavms_SLQSAVMSGetSettings_t	955
unpack_swiaavms_SLQSAVMSGetSettings_v2_t	958
unpack_swiaavms_SLQSAVMSSendSelection_t	960
unpack_swiaavms_SLQSAVMSSessionGetInfo_t	961
unpack_swiaavms_SLQSAvmsSetEventReport_t	962
unpack_swiaavms_SLQSAVMSSetSettings_t	962
unpack_swiaavms_SLQSAVMSSetSettings_v2_t	963
unpack_swiaavms_SLQSAVMSStartSession_t	963
unpack_swiaavms_SLQSAVMSStopSession_t	964
unpack_swidms_SLQSSwiDmsGetHWWatchdog_t	964
unpack_swidms_SLQSSwiDmsGetMTU_t	965
unpack_swidms_SLQSSwiDmsGetSecureInfo_t	965
unpack_swidms_SLQSSwiDmsGetUsbComp_t	966
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t	967
unpack_swidms_SLQSSwiDmsSetHWWatchdog_t	968
unpack_swidms_SLQSSwiDmsSetMTU_t	968
unpack_swidms_SLQSSwiDmsSetUsbComp_t	969
unpack_swiloc_SwiLocGetAutoStart_t	969
unpack_swioma_SLQSOMADMAAlertCallback_ind_t	971
unpack_swioma_SLQSOMADMGetSessionInfo_t	972
unpack_swioma_SLQSOMADMGetSessionInfoExt_t	975
unpack_swioma_SLQSOMADMGetSettings_t	979
unpack_swioma_SLQSOMADMStartSession_t	980
unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t	981
unpack_tmd_SLQSTmdGetMitigationDevList_t	981
unpack_tmd_SLQSTmdGetMitigationLvl_t	982
unpack_tmd_SLQSTmdMitigationLvlRptCallback_ind_t	983
unpack_tmd_SLQSTmdRegNotMitigationLvl_t	983
unpack_uim_ChangePin_t	984
unpack_uim_GetCardStatus_t	985
unpack_uim_GetCardStatusV2_t	986
unpack_uim_ReadTransparent_t	987
unpack_uim_SetPinProtection_t	988
unpack_uim_SetUimSlotStatusChangeCallback_ind_t	988
unpack_uim_SLQSUIMAuthenticate_t	989
unpack_uim_SLQSUIMDepersonalization_t	990
unpack_uim_SLQSUIMEventRegister_t	990
unpack_uim_SLQSUIMGetConfiguration_t	991
unpack_uim_SLQSUIMGetFileAttributes_t	992
unpack_uim_SLQSUIMGetServiceStatus_t	993
unpack_uim_SLQSUIMGetSlotsStatus_t	994
unpack_uim_SLQSUIMGetSlotsStatusV2_t	994
unpack_uim_SLQSUIMReadRecord_t	995
unpack_uim_SLQSUIMRefreshCallback_Ind_t	996
unpack_uim_SLQSUIMRefreshGetLastEvent_t	997
unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t	997
unpack_uim_SLQSUIMWriteRecord_t	998
unpack_uim_SLQSUIMWriteTransparent_t	998

unpack_uim_UnblockPin_t	999
unpack_uim_UnblockPinV2_t	1000
unpack_uim_VerifyPin_t	1001
unpack_voice_allCallStatusCallback_ind_t	1002
unpack_voice_DTMFEventCallback_ind_t	1004
unpack_voice_OTASPStatusCallback_ind_t	1005
unpack_voice_SLQSPoriginateUSSD_t	1006
unpack_voice_SLQSVoiceAnswerCall_t	1009
unpack_voice_SLQSVoiceBurstDTMF_t	1010
unpack_voice_SLQSVoiceDialCall_t	1010
unpack_voice_SLQSVoiceEndCall_t	1011
unpack_voice_SLQSVoiceGetAllCallInfo_t	1012
unpack_voice_SLQSVoiceGetCallBarring_t	1016
unpack_voice_SLQSVoiceGetCallForwardingStatus_t	1017
unpack_voice_SLQSVoiceGetCallInfo_t	1019
unpack_voice_SLQSVoiceGetCallWaiting_t	1022
unpack_voice_SLQSVoiceGetCLIP_t	1023
unpack_voice_SLQSVoiceGetCLIR_t	1025
unpack_voice_SLQSVoiceGetCNAP_t	1026
unpack_voice_SLQSVoiceGetCOLP_t	1028
unpack_voice_SLQSVoiceGetCOLR_t	1029
unpack_voice_SLQSVoiceGetConfig_t	1030
unpack_voice_SLQSVoiceManageCalls_t	1032
unpack_voice_SLQSVoiceSendFlash_t	1033
unpack_voice_SLQSVoiceSetCallBarringPassword_t	1034
unpack_voice_SLQSVoiceSetConfig_t	1035
unpack_voice_SLQSVoiceSetSUPSService_t	1037
unpack_voice_SLQSVoiceStartContDTMF_t	1038
unpack_voice_SLQSVoiceStopContDTMF_t	1038
unpack_voice_SLQSVoiceSUPSCallback_ind_t	1039
unpack_voice_SUPSNotificationCallback_ind_t	1042
unpack_voice_USSDNotificationCallback_ind_t	1044
unpack_voice_VoiceInfoRecCallback_ind_t	1044
unpack_voice_voicePrivacyChangeCallback_ind_t	1047
unpack_wds_DHCPv4ClientLease_ind_t	1048
unpack_wds_GetAutoconnect_t	1049
unpack_wds_GetByteTotals_t	1049
unpack_wds_GetConnectionRate_t	1050
unpack_wds_GetDataBearerTechnology_t	1051
unpack_wds_GetDefaultProfile_t	1052
unpack_wds_GetDefaultProfileNum_t	1055
unpack_wds_GetDefaultProfileV2_t	1055
unpack_wds_GetDormancyState_t	1058
unpack_wds_GetLastMobileIPError_t	1059
unpack_wds_GetMobileIP_t	1060
unpack_wds_GetMobileIPProfile_t	1060
unpack_wds_GetPacketStatistics_t	1062
unpack_wds_GetPacketStatus_t	1064
unpack_wds_GetSessionDuration_t	1066
unpack_wds_GetSessionDurationV2_t	1067
unpack_wds_GetSessionState_t	1068
unpack_wds_RMSetTransferStatistics_t	1068
unpack_wds_SetMobileIPProfile_t	1069
unpack_wds_SLQSCreateProfile_t	1069
unpack_wds_SLQSDeleteProfile_t	1070
unpack_wds_SLQSDUNCallInfoCallBack_ind_t	1070
unpack_wds_SLQSGet3GPPConfigItem_t	1072
unpack_wds_SLQSGetCurrDataSystemStat_t	1074

unpack_wds_SLQSGetCurrentChannelRate_t	1075
unpack_wds_SLQSGetDataBearerTechnology_t	1076
unpack_wds_SLQSGetDUNCallInfo_t	1077
unpack_wds_SLQSGetProfileSettings_t	1080
unpack_wds_SLQSGetProfileSettingsV2_t	1080
unpack_wds_SLQSGetRuntimeSettings_t	1081
unpack_wds_SLQSModifyProfile_t	1084
unpack_wds_SLQSSetIPFamilyPreference_t	1085
unpack_wds_SLQSSetPacketSrvStatusCallback_t	1085
unpack_wds_SLQSSetWdsEventCallback_ind_t	1087
unpack_wds_SLQSSetDHCPv4ClientConfig_t	1093
unpack_wds_SLQSSetLoopback_t	1094
unpack_wds_SLQSStartDataSession_t	1095
unpack_wds_SLQSSwiProfileChangeCallback_Ind_t	1096
unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t	1096
UnPackGetProfileSettingOut	1099
UnPackGetProfileSettingOutV2	1099
UnpackSwiAvmsEventReportBinaryUpdateSessionInfo	1100
UnpackSwiAvmsEventReportConfig	1102
UnpackSwiAvmsEventReportConnectionRequest	1103
UnpackSwiAvmsEventReportDataSessionStatus	1103
UnpackSwiAvmsEventReportHTTPStatus	1104
UnpackSwiAvmsEventReportNotification	1106
UnpackSwiAvmsEventReportPackageID	1107
UnpackSwiAvmsEventReportRegStatus	1107
UnpackSwiAvmsEventReportSessionType	1108
UnpackSwiAvmsEventReportWAMSPParamChange	1108
unpackWdsProfileParam	1109
unpackWdsProfileParamV2	1109
voice_airTimer	1110
voice_allCallsAlphaIDInfo	1110
voice_allCallsDiagInfo	1111
voice_allCallsUUSInfo	1111
voice_alphaIDInfo	1112
voice_arrAlertingPattern	1112
voice_arrAlertingType	1113
voice_arrAlphaID	1114
voice_arrCalledPartyNum	1114
voice_arrCallEndReason	1115
voice_arrCallInfo	1116
voice_arrConnectPartyNum	1116
voice_arrDiagInfo	1117
voice_arrRedirPartyNum	1117
voice_arrRemotePartyName	1118
voice_arrRemotePartyNum	1118
voice_arrSvcOption	1119
voice_arrUUSInfo	1120
voice_burstDTMFInfo	1120
voice_calledPartyInfo	1121
voice_calledPartySubAdd	1122
voice_callerIDInfo	1123
voice_callFwdTypeAndPlan	1124
voice_callFWExtInfo	1125
voice_callFWInfo	1127
voice_callInfo	1128
voice_callingPartyInfo	1129
voice_ccSUPSType	1131
voice_CLIPResp	1132

voice_CLIRResp	1132
voice_CNAPResp	1133
voice_COLPResp	1134
voice_COLRResp	1134
voice_connectNumInfo	1135
voice_CUGInfo	1137
voice_curAMRConfig	1137
voice_diagInfo	1138
voice_DTMFInfo	1139
voice_DTMFLengths	1139
voice_ECTNum	1140
voice_extDispRecInfo	1141
voice_getAllCallInformation	1142
voice_getAllCallRmtPtyName	1142
voice_getAllCallRmtPtyNum	1143
voice_getCallFWExtInfo	1143
voice_getCallFWInfo	1144
voice_lineCtrlInfo	1144
voice_newPwdData	1145
voice_NSSAudioCtrl	1146
voice_peerNumberInfo	1146
voice_prefVoiceSO	1148
voice_redirNumInfo	1150
voice_remotePartyName	1152
voice_remotePartyNum	1152
voice_roamTimer	1153
voice_signalInfo	1154
voice_SUPSInfo	1155
voice_USSDNotificationNetworkInfo	1155
voice_USSInfo	1156
voice_UUSInfo	1156
wds_channelRate	1158
wds_ChannelRateTlv	1158
wds_ConnStatusTlv	1159
wds_currNetworkInfo	1159
wds_DataBearTechTlv	1160
wds_DataULongLongTlv	1161
wds_DataULongTlv	1162
wds_DHCPLeaseOptTlv	1162
wds_DHCPLeaseStateTlv	1163
wds_DHCPOpt	1163
wds_DHCPProfileIdTlv	1164
wds_DHCPv4HWConfig	1164
wds_DHCPv4Option	1165
wds_DHCPv4OptionList	1166
wds_DHCPv4ProfileId	1166
wds_Domain	1167
wds_DomainNameList	1167
wds_DormStatTlv	1168
wds_GPRSQoS	1168
wds_IPv4AdTlv	1169
wds_IPV6AddressInfo	1170
wds_IPV6GWAddressInfo	1170
wds_LastMdmCallEndRsnTlv	1171
wds_PCSCFFQDNAddress	1171
wds_PCSCFFQDNAddressList	1172
wds_PCSCFIPv4ServerAddressList	1172
wds_profileChange	1173

wds_ProfileIdentifier	1173
wds_profileInfo	1174
wds_RXBytesOKTlv	1175
wds_sourceOfChange	1175
wds_transferStatInd	1175
wds_TrStatInd	1176
wds_TXBytesOKTlv	1177
wds_UMTSMInQoS	1177
wdsDhcpv4HwConfig	1179
wdsDhcpv4Option	1180
wdsDhcpv4OptionList	1181
wdsDhcpv4ProfileId	1181



## 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	1183
<a href="#">audio.h</a>		1183
<a href="#">cat.h</a>		1189
<a href="#">common.h</a>		1192
<a href="#">dms.h</a>		1198
<a href="#">fms.h</a>		1260
<a href="#">ims.h</a>		1263
<a href="#">imsa.h</a>		1275
<a href="#">lite-fw.h</a>		1281
<a href="#">loc.h</a>		1294
<a href="#">nas.h</a>		1319
<a href="#">pds.h</a>		1387
<a href="#">qaGobiApiTableBandClasses.h</a>	Network Access Service API Band Classes table	1409
<a href="#">qaGobiApiTableCallControlReturnReasons.h</a>	Call Control Return Reasons table	1413
<a href="#">qaGobiApiTableCallEndReasons.h</a>	Wireless Data Service Call End Reasons	1414
<a href="#">qaGobiApiTableCarrierCodes.h</a>	Carrier Codes table	1430
<a href="#">qaGobiApiTableCodingScheme.h</a>	Data Coding Scheme	1432
<a href="#">qaGobiApiTableGpsCapabilityCodes.h</a>	Position Determination Service API GPS Capability Codes	1435
<a href="#">qaGobiApiTablePowerModes.h</a>	Device Management Service API Power Modes table	1435
<a href="#">qaGobiApiTableRadioInterfaces.h</a>	Network Access Service API Radio Interfaces table	1436
<a href="#">qaGobiApiTableRegionCodes.h</a>	Region Codes table	1436
<a href="#">qaGobiApiTableServiceOptions.h</a>	Voice Service Options	1437
<a href="#">qaGobiApiTableSupServiceInfoClasses.h</a>	Voice Supplementary Service Information Classes	1439
<a href="#">qaGobiApiTableSwiAudio.h</a>	Swi Audio related tables	1440

<a href="#">qaGobiApiTableSwiOMADMSessionStatus.h</a>	
Session Status table	1440
<a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a>	
Update Complete Status table	1441
<a href="#">qaGobiApiTableVoiceCallEndReasons.h</a>	
Voice Service Call and supplementary services end reasons	1443
<a href="#">qmerrno.h</a>	1449
<a href="#">qos.h</a>	1456
<a href="#">rms.h</a>	1466
<a href="#">sar.h</a>	1468
<a href="#">sms.h</a>	1470
<a href="#">swiaudio.h</a>	1492
<a href="#">swiavms.h</a>	1502
<a href="#">SwiDataTypes.h</a>	
SWI data types	1516
<a href="#">swidms.h</a>	1517
<a href="#">swiloc.h</a>	1525
<a href="#">swioma.h</a>	1527
<a href="#">swiomaext.h</a>	1537
<a href="#">switype_256bit.h</a>	1543
<a href="#">SWIWWANCMAPI.h</a>	1545
<a href="#">tmd.h</a>	1545
<a href="#">uim.h</a>	1549
<a href="#">voice.h</a>	1574
<a href="#">wds.h</a>	1606

## Chapter 6

# Module Documentation

### 6.1 QMI pack/unpack (lite-qmi)

#### Files

- file [common.h](#)
- file [dms.h](#)
- file [loc.h](#)
- file [nas.h](#)
- file [qos.h](#)
- file [sms.h](#)
- file [swioma.h](#)
- file [uim.h](#)
- file [wds.h](#)
- file [fms.h](#)
- file [swiloc.h](#)
- file [pds.h](#)
- file [rms.h](#)
- file [sar.h](#)
- file [swiavms.h](#)
- file [voice.h](#)
- file [cat.h](#)
- file [imsa.h](#)
- file [tmd.h](#)
- file [swiomaext.h](#)
- file [audio.h](#)
- file [swiaudio.h](#)
- file [ims.h](#)
- file [swidms.h](#)

#### 6.1.1 Detailed Description

## 6.2 Streaming Download Protocol (lite-fw)

### Files

- file [lite-fw.h](#)

### 6.2.1 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 \_litefw\_FirmwareFileInfo Struct Reference

#### Data Fields

- char [fullPath](#) [512]
- uint8\_t [imageMask](#)
- enum [litefw\\_imagetype](#) [headerType](#)
- enum [litefw\\_fileimgtype](#) [imageType](#)
- char [modelIdStr](#) [20+1]
- char [partNoStr](#) [20+1]
- char [skuStr](#) [20+1]
- char [packageIdStr](#) [20+1]
- char [carrierStr](#) [20+1]
- char [priVersionStr](#) [20+1]
- char [versionStr](#) [84]
- char [releaseDate](#) [8]

#### 8.1.1 Detailed Description

This structure provided more detailed information of the provided firmware file

#### Parameters

<i>fullPath</i>	<ul style="list-style-type: none"><li>• full path of the file</li></ul>
<i>imagemask</i>	<ul style="list-style-type: none"><li>• Bitmask provides type of file<ul style="list-style-type: none"><li>– bit0 - cwe</li><li>– bit1 - nvu</li><li>– bit2 - spk</li></ul></li></ul>
<i>headerType</i>	<ul style="list-style-type: none"><li>• see <a href="#">litefw_imagetype</a></li></ul>
<i>imageType</i>	<ul style="list-style-type: none"><li>• see <a href="#">litefw_fileimgtype</a></li></ul>

<i>modelIdStr</i>	<ul style="list-style-type: none"> <li>• device model id</li> </ul>
<i>partNoStr</i>	<ul style="list-style-type: none"> <li>• part number id</li> </ul>
<i>skuStr</i>	<ul style="list-style-type: none"> <li>• sku id</li> </ul>
<i>packageIdStr</i>	<ul style="list-style-type: none"> <li>• cwe sierra package id</li> </ul>
<i>carrierStr</i>	<ul style="list-style-type: none"> <li>• carrier id</li> </ul>
<i>priVersionStr</i>	<ul style="list-style-type: none"> <li>• pri version</li> </ul>
<i>versionStr</i>	<ul style="list-style-type: none"> <li>• firmware version</li> </ul>
<i>releaseDate</i>	<ul style="list-style-type: none"> <li>• release date of the file</li> </ul>

## 8.1.2 Field Documentation

8.1.2.1 `char _litefw_FirmwareFileInfo::carrierStr[20+1]`

8.1.2.2 `char _litefw_FirmwareFileInfo::fullPath[512]`

8.1.2.3 `enum litefw_imgetype _litefw_FirmwareFileInfo::headerType`

8.1.2.4 `uint8_t _litefw_FirmwareFileInfo::imageMask`

8.1.2.5 `enum litefw_fileimgtype _litefw_FirmwareFileInfo::imageType`

8.1.2.6 `char _litefw_FirmwareFileInfo::modelIdStr[20+1]`

8.1.2.7 `char _litefw_FirmwareFileInfo::packageIdStr[20+1]`

8.1.2.8 `char _litefw_FirmwareFileInfo::partNoStr[20+1]`

8.1.2.9 `char _litefw_FirmwareFileInfo::priVersionStr[20+1]`

8.1.2.10 `char _litefw_FirmwareFileInfo::releaseDate[8]`

8.1.2.11 `char _litefw_FirmwareFileInfo::skuStr[20+1]`

8.1.2.12 `char _litefw_FirmwareFileInfo::versionStr[84]`

## 8.2 `_litefw_FirmwareInfo_` Struct Reference



## Data Fields

- char [szModelid\\_str](#) [20+1]
- char [szFwversion\\_str](#) [20+1]
- char [szSku\\_str](#) [20+1]
- char [szPackageid\\_str](#) [20+1]
- char [szCarrier\\_str](#) [20+1]
- char [szCarrierPriversion\\_str](#) [20+1]

### 8.2.1 Detailed Description

This structure contains information of the provided firmware file

#### Parameters

<i>szModelid_str</i>	<ul style="list-style-type: none"> <li>• Model Name String</li> </ul>
<i>szFwversion_str</i>	<ul style="list-style-type: none"> <li>• Firmware Version String.</li> </ul>
<i>szSku_str</i>	<ul style="list-style-type: none"> <li>• SKU String.</li> </ul>
<i>szPackageid_str</i>	<ul style="list-style-type: none"> <li>• Package ID String.</li> </ul>
<i>szCarrier_str</i>	<ul style="list-style-type: none"> <li>• Carrier String.</li> </ul>
<i>szCarrier-Priversion_str</i>	<ul style="list-style-type: none"> <li>• Carrier PRI Version String.</li> </ul>

### 8.2.2 Field Documentation

8.2.2.1 char \_litefw\_FirmwareInfo::szCarrier\_str[20+1]

8.2.2.2 char \_litefw\_FirmwareInfo::szCarrierPriversion\_str[20+1]

8.2.2.3 char \_litefw\_FirmwareInfo::szFwversion\_str[20+1]

8.2.2.4 char \_litefw\_FirmwareInfo::szModelid\_str[20+1]

8.2.2.5 char \_litefw\_FirmwareInfo::szPackageid\_str[20+1]

8.2.2.6 char \_litefw\_FirmwareInfo::szSku\_str[20+1]

## 8.3 \_litefw\_FirmwarePartNo\_ Struct Reference

## Data Fields

- char [szPartno\\_str](#) [20+1]

### 8.3.1 Detailed Description

This structure contains information of the response parameters associated with a Read Transparent API.

#### Parameters

<i>szPartno_str</i>	<ul style="list-style-type: none"> <li>Part Number of the provided firmware image</li> </ul>
---------------------	--

### 8.3.2 Field Documentation

8.3.2.1 `char _litefw_FirmwarePartNo_::szPartno_str[20+1]`

## 8.4 altSrcInfo\_t Struct Reference

### Data Fields

- `uint32_t` [source](#)
- `uint32_t` [linkage](#)
- `uint32_t` [coverage](#)

### 8.4.1 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>
-----------------	---

## 8.4.2 Field Documentation

8.4.2.1 `uint32_t altSrcInfo_t::coverage`

8.4.2.2 `uint32_t altSrcInfo_t::linkage`

8.4.2.3 `uint32_t altSrcInfo_t::source`

## 8.5 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`

### 8.5.1 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>

## 8.5.2 Field Documentation

8.5.2.1 `uint8_t appStats::aidLength`

8.5.2.2 `uint8_t appStats::aidVal[255]`

8.5.2.3 `uint8_t appStats::appState`

8.5.2.4 `uint8_t appStats::appType`

8.5.2.5 `uint8_t appStats::persoFeature`

8.5.2.6 `uint8_t appStats::persoRetries`

8.5.2.7 `uint8_t appStats::persoState`

8.5.2.8 uint8\_t appStats::persoUnblockRetries

8.5.2.9 uint8\_t appStats::pin1Retries

8.5.2.10 uint8\_t appStats::pin1State

8.5.2.11 uint8\_t appStats::pin2Retries

8.5.2.12 uint8\_t appStats::pin2State

8.5.2.13 uint8\_t appStats::puk1Retries

8.5.2.14 uint8\_t appStats::puk2Retries

8.5.2.15 uint8\_t appStats::univPin

## 8.6 audio\_RXAGCList Struct Reference

### Data Fields

- uint16\_t \* [pRXStaticGain](#)
- uint16\_t \* [pRXAIG](#)
- uint16\_t \* [pRXExpThres](#)
- uint16\_t \* [pRXExpSlope](#)
- uint16\_t \* [pRXComprThres](#)
- uint16\_t \* [pRXComprSlope](#)

### 8.6.1 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>

### 8.6.2 Field Documentation

- 8.6.2.1 uint16\_t\* audio\_RXAGCList::pRXAIG
- 8.6.2.2 uint16\_t\* audio\_RXAGCList::pRXComprSlope
- 8.6.2.3 uint16\_t\* audio\_RXAGCList::pRXComprThres
- 8.6.2.4 uint16\_t\* audio\_RXAGCList::pRXExpSlope
- 8.6.2.5 uint16\_t\* audio\_RXAGCList::pRXExpThres
- 8.6.2.6 uint16\_t\* audio\_RXAGCList::pRXStaticGain

## 8.7 audio\_RXAVCList Struct Reference

### Data Fields

- uint16\_t \* [pAVRXAVCSens](#)
- uint16\_t \* [pAVRXAVCHheadroom](#)

### 8.7.1 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>

### 8.7.2 Field Documentation

- 8.7.2.1 uint16\_t\* audio\_RXAVCList::pAVRXAVCHheadroom
- 8.7.2.2 uint16\_t\* audio\_RXAVCList::pAVRXAVCSens

## 8.8 audio\_RXPCMIRFitr Struct Reference

### Data Fields

- uint16\_t \* [pFlag](#)
- uint16\_t \* [pStageCnt](#)
- uint8\_t \* [pStage0Val](#)
- uint8\_t \* [pStage1Val](#)
- uint8\_t \* [pStage2Val](#)
- uint8\_t \* [pStage3Val](#)
- uint8\_t \* [pStage4Val](#)

### 8.8.1 Detailed Description

This structure contains the SLQSGetAudioPathConfig parameters related to AV\_RXPCMIRFLTR.

## 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>



## 8.8.2 Field Documentation

8.8.2.1 uint16\_t\* audio\_RXPCMIIRFiltr::pFlag

8.8.2.2 uint8\_t\* audio\_RXPCMIIRFiltr::pStage0Val

8.8.2.3 uint8\_t\* audio\_RXPCMIIRFiltr::pStage1Val

8.8.2.4 uint8\_t\* audio\_RXPCMIIRFiltr::pStage2Val

8.8.2.5 uint8\_t\* audio\_RXPCMIIRFiltr::pStage3Val

8.8.2.6 uint8\_t\* audio\_RXPCMIIRFiltr::pStage4Val

8.8.2.7 uint16\_t\* audio\_RXPCMIIRFiltr::pStageCnt

## 8.9 audio\_TXAGCList Struct Reference

### Data Fields

- uint16\_t \* [pTXStaticGain](#)
- uint16\_t \* [pTXAIG](#)
- uint16\_t \* [pTXExpThres](#)
- uint16\_t \* [pTXExpSlope](#)
- uint16\_t \* [pTXComprThres](#)
- uint16\_t \* [pTXComprSlope](#)

### 8.9.1 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>

### 8.9.2 Field Documentation

8.9.2.1 uint16\_t\* audio\_TXAGCList::pTXAIG

8.9.2.2 uint16\_t\* audio\_TXAGCList::pTXComprSlope

8.9.2.3 uint16\_t\* audio\_TXAGCList::pTXComprThres

8.9.2.4 uint16\_t\* audio\_TXAGCList::pTXExpSlope

8.9.2.5 uint16\_t\* audio\_TXAGCList::pTXExpThres

8.9.2.6 uint16\_t\* audio\_TXAGCList::pTXStaticGain

## 8.10 audio\_TXPCMIIRFitr Struct Reference

### Data Fields

- uint16\_t \* [pFlag](#)
- uint16\_t \* [pStageCnt](#)
- uint8\_t \* [pStage0Val](#)
- uint8\_t \* [pStage1Val](#)
- uint8\_t \* [pStage2Val](#)
- uint8\_t \* [pStage3Val](#)
- uint8\_t \* [pStage4Val](#)

### 8.10.1 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>

## 8.10.2 Field Documentation

8.10.2.1 uint16\_t\* audio\_TXPCMIIRFitr::pFlag

8.10.2.2 uint8\_t\* audio\_TXPCMIIRFitr::pStage0Val

8.10.2.3 uint8\_t\* audio\_TXPCMIIRFitr::pStage1Val

8.10.2.4 uint8\_t\* audio\_TXPCMIIRFitr::pStage2Val

8.10.2.5 uint8\_t\* audio\_TXPCMIIRFitr::pStage3Val

8.10.2.6 uint8\_t\* audio\_TXPCMIIRFitr::pStage4Val

8.10.2.7 uint16\_t\* audio\_TXPCMIIRFitr::pStageCnt

## 8.11 CarrierImage\_t Struct Reference

## Data Fields

- uint32\_t [m\\_nCarrierId](#)
- uint32\_t [m\\_nFolderId](#)
- uint32\_t [m\\_nStorage](#)
- uint8\_t [m\\_FwImageId](#) [100]
- uint8\_t [m\\_FwBuildId](#) [100]
- uint8\_t [m\\_PriImageId](#) [100]
- uint8\_t [m\\_PriBuildId](#) [100]

### 8.11.1 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>

### 8.11.2 Field Documentation

8.11.2.1 uint8\_t CarrierImage\_t::m\_FwBuildId[100]

8.11.2.2 uint8\_t CarrierImage\_t::m\_FwImageId[100]

8.11.2.3 uint32\_t CarrierImage\_t::m\_nCarrierId

8.11.2.4 uint32\_t CarrierImage\_t::m\_nFolderId

8.11.2.5 uint32\_t CarrierImage\_t::m\_nStorage

8.11.2.6 `uint8_t CarrierImage_t::m_PriBuildId[100]`

8.11.2.7 `uint8_t CarrierImage_t::m_PrImageld[100]`

## 8.12 cat\_AlphaIdentifierTlv Struct Reference

### Data Fields

- `uint8_t ReferenceID`
- `uint16_t AlphaDLength`
- `uint8_t AlphaID [255]`

### 8.12.1 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]

### 8.12.2 Field Documentation

8.12.2.1 `uint8_t cat_AlphaIdentifierTlv::AlphaID[255]`

8.12.2.2 `uint16_t cat_AlphaIdentifierTlv::AlphaDLength`

8.12.2.3 `uint8_t cat_AlphaIdentifierTlv::ReferenceID`

## 8.13 cat\_commonEventTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t EventID`
- `uint16_t EventLength`
- union `cat_currentCatEvent CatEvent`

### 8.13.1 Detailed Description

Structure used to store all Common CAT Event TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>EventID</i>	- Event ID. Can be any of the following 16 - Display Text 17 - Get In-Key 18 - Get Input 19 - Setup Menu 20 - Select Item 21 - Send SMS - Alpha Identifier 22 - Setup Event List 23 - Setup Idle Mode Text 24 - Language Notification 25 - Refresh 26 - End Proactive Session
<i>EventLength</i>	- Length of pData ( in Bytes )
<i>CatEvent</i>	- Structure to the Data specific to the CAT event ID

### 8.13.2 Field Documentation

8.13.2.1 union `cat_currentCatEvent` `cat_commonEventTlv::CatEvent`

8.13.2.2 `uint8_t` `cat_commonEventTlv::EventID`

8.13.2.3 `uint16_t` `cat_commonEventTlv::EventLength`

8.13.2.4 `uint8_t` `cat_commonEventTlv::TlvPresent`

## 8.14 `cat_currentCatEvent` Union Reference

### Data Fields

- struct [cat\\_EventIDDataTlv](#) `CatEvIDData`
- struct [cat\\_AIPhaldentifierTlv](#) `CatAlphaldtfr`
- struct [cat\\_EventListTlv](#) `CatEventLst`
- struct [cat\\_RefreshTlv](#) `CatRefresh`
- struct [cat\\_EndProactiveSessionTlv](#) `CatEndPS`

### 8.14.1 Detailed Description

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

- Use [cat\\_EventIDDataTlv](#) if the Event ID is any of the below.
  - 16
  - 17
  - 18
  - 19
  - 20
  - 23
  - 24
- Use [cat\\_AIPhaldentifierTlv](#) if the Event ID is 21
- Use [cat\\_EventListTlv](#) if the Event ID is 22
- Use [cat\\_RefreshTlv](#) if the Event ID is 25
- Use [cat\\_EndProactiveSessionTlv](#) if the Event ID is 26

### 8.14.2 Field Documentation

8.14.2.1 struct `cat_AIPhaldentifierTlv` `cat_currentCatEvent::CatAlphaldtfr`

8.14.2.2 struct `cat_EndProactiveSessionTlv` `cat_currentCatEvent::CatEndPS`

8.14.2.3 struct `cat_EventListTlv` `cat_currentCatEvent::CatEventLst`

8.14.2.4 struct `cat_EventIDDataTlv` `cat_currentCatEvent::CatEvIDData`

8.14.2.5 struct `cat_RefreshTlv` `cat_currentCatEvent::CatRefresh`

## 8.15 cat\_EndProactiveSessionTlv Struct Reference

### Data Fields

- uint8\_t [EndProactiveSession](#)

### 8.15.1 Detailed Description

structure used to store End Proactive Session event parameters.

#### Parameters

<i>EndProactiveSession</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>
----------------------------	--

### 8.15.2 Field Documentation

8.15.2.1 uint8\_t cat\_EndProactiveSessionTlv::EndProactiveSession

## 8.16 cat\_EventIDDataTlv Struct Reference

### Data Fields

- uint32\_t [ReferenceID](#)
- uint16\_t [DataLength](#)
- uint8\_t [Data](#) [255]

### 8.16.1 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]

### 8.16.2 Field Documentation

8.16.2.1 uint8\_t cat\_EventIDDataTlv::Data[255]

8.16.2.2 uint16\_t cat\_EventIDDataTlv::DataLength

8.16.2.3 uint32\_t cat\_EventIDDataTlv::ReferenceID

## 8.17 cat\_EventListTlv Struct Reference

### Data Fields

- uint32\_t [SetupEventList](#)

### 8.17.1 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>
-----------------------	---

### 8.17.2 Field Documentation

8.17.2.1 `uint32_t cat_EventListTlv::SetupEventList`

## 8.18 cat\_RefreshTlv Struct Reference

#### Data Fields

- `uint16_t` [RefreshMode](#)
- `uint8_t` [RefreshStage](#)

### 8.18.1 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>

### 8.18.2 Field Documentation

8.18.2.1 `uint16_t cat_RefreshTlv::RefreshMode`

8.18.2.2 `uint8_t cat_RefreshTlv::RefreshStage`

## 8.19 cdmaSSInfo Struct Reference

#### Data Fields

- `int8_t` [rssi](#)
- `int16_t` [ecio](#)



### 8.19.1 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>

### 8.19.2 Field Documentation

8.19.2.1 `int16_t cdmaSSInfo::ecio`

8.19.2.2 `int8_t cdmaSSInfo::rssi`

## 8.20 connectionStatus Struct Reference

#### Data Fields

- `uint8_t` [MDMConnStatus](#)
- `uint64_t` [MDMCallDuration](#)

### 8.20.1 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> <li>– 0xff - Invalid data.</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). <ul style="list-style-type: none"> <li>– 0xffffffffffffff - Invalid data.</li> </ul> </li> </ul>

### 8.20.2 Field Documentation

8.20.2.1 `uint64_t connectionStatus::MDMCallDuration`

8.20.2.2 `uint8_t` `connectionStatus::MDMConnStatus`

## 8.21 crashInfoParams Struct Reference

### Data Fields

- `uint8_t` [crashStatus](#)
- [crashInformation](#) `crashInfo`

### 8.21.1 Detailed Description

This structure contains crash information parameters

#### Parameters

<i>crashStatus[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>crashInfo[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">crashInformation</a></li> </ul>

### 8.21.2 Field Documentation

8.21.2.1 `crashInformation` `crashInfoParams::crashInfo`

8.21.2.2 `uint8_t` `crashInfoParams::crashStatus`

## 8.22 crashInformation Struct Reference

### Data Fields

- `uint16_t` [numCrashes](#)
- `uint32_t` [crashId](#)
- `uint32_t` [crashData](#)
- `uint16_t` [crashStrlen](#)
- `char` [crashString](#) [255]
- `uint16_t` [gcdumpStrlen](#)
- `char` [gcdumpString](#) [1024]

### 8.22.1 Detailed Description

This structure contains crash information

#### Parameters

<i>numCrashes[OUT]</i>	<ul style="list-style-type: none"> <li>• number of instances of the remaining fields</li> </ul>
<i>crashId[OUT]</i>	<ul style="list-style-type: none"> <li>• random crash id assigned at crash</li> </ul>

<i>crashData</i>	<ul style="list-style-type: none"> <li>• crash data[OUT]</li> </ul>
<i>crashStrlen</i> [IN/-OUT]	<ul style="list-style-type: none"> <li>• length of the crashString field as an input, length of the crashString field returned by API as an OUTPUT</li> </ul>
<i>crashString</i>	<ul style="list-style-type: none"> <li>• crash string</li> </ul>
<i>gcdumpStrlen</i> [O-UT]	<ul style="list-style-type: none"> <li>• length of the gcdumpString field as an input, length of the gcdumpString field returned by API as an OUTPUT</li> </ul>
<i>gcdumpString</i> [O-UT]	<ul style="list-style-type: none"> <li>• gcdump string for the crash</li> </ul>

## 8.22.2 Field Documentation

8.22.2.1 `uint32_t` `crashInformation::crashData`

8.22.2.2 `uint32_t` `crashInformation::crashId`

8.22.2.3 `char` `crashInformation::crashString[255]`

8.22.2.4 `uint16_t` `crashInformation::crashStrlen`

8.22.2.5 `char` `crashInformation::gcdumpString[1024]`

8.22.2.6 `uint16_t` `crashInformation::gcdumpStrlen`

8.22.2.7 `uint16_t` `crashInformation::numCrashes`

## 8.23 currNetworkInfo Struct Reference

### Data Fields

- `uint8_t` [NetworkType](#)
- `uint32_t` [RATMask](#)
- `uint32_t` [SOMask](#)

### 8.23.1 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>	<a href="#">RAT Mask</a>
<i>SOMask</i>	<a href="#">SO Mask</a>

### 8.23.2 Field Documentation

8.23.2.1 `uint8_t currNetworkInfo::NetworkType`

8.23.2.2 `uint32_t currNetworkInfo::RATMask`

8.23.2.3 `uint32_t currNetworkInfo::SOMask`

## 8.24 dms\_ActivationStatusTlv Struct Reference

### Data Fields

- `uint16_t` [TlvPresent](#)
- `uint32_t` [activationStatus](#)

### 8.24.1 Detailed Description

Activation Status Tlv

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<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>

### 8.24.2 Field Documentation

8.24.2.1 `uint32_t dms_ActivationStatusTlv::activationStatus`

8.24.2.2 `uint16_t dms_ActivationStatusTlv::TlvPresent`

## 8.25 dms\_devCaps Struct Reference

## Data Fields

- uint32\_t [MaxTXChannelRate](#)
- uint32\_t [MaxRXChannelRate](#)
- uint8\_t [DataServiceCapability](#)
- uint8\_t [SimCapability](#)
- uint8\_t [RadiolfacesSize](#)
- uint8\_t [Radiolfaces](#) [255]

### 8.25.1 Detailed Description

This structure contains the [dms\\_devCaps](#) parameters.

#### Parameters

<i>MaxTXChannelRate</i>	<ul style="list-style-type: none"> <li>• Maximum Tx channel rate in bits per second</li> </ul>
<i>MaxRXChannelRate</i>	<ul style="list-style-type: none"> <li>• Maximum Rx channel rate in bits per second</li> </ul>
<i>DataServiceCapability</i>	<ul style="list-style-type: none"> <li>• DMS_DATA_CAP_NONE - 0x00</li> <li>• DMS_DATA_CAP_CS_ONLY-circuit switched only - 0x01</li> <li>• DMS_DATA_CAP_PS_ONLY-packet switched only - 0x02</li> <li>• DMS_DATA_CAP_SIMUL_CS_AND_PS - 0x03</li> <li>• DMS_DATA_CAP_NONSIMUL_CS_AND_PS - 0x04</li> </ul>
<i>SimCapability</i>	<ul style="list-style-type: none"> <li>• SIM_NOT_SUPPORTED - 0x01</li> <li>• SIM_SUPPORTED - 0x02</li> </ul>
<i>RadiolfacesSize</i>	-Number of radio interfaces
<i>Radiolfaces</i>	-List of radio interfaces, each byte will be one of following <ul style="list-style-type: none"> <li>• DMS_RADIO_IF_1X- CDMA2000 1X - 0x01</li> <li>• DMS_RADIO_IF_1X_EVDO- CDMA2000 HRPD - 0x02</li> <li>• DMS_RADIO_IF_GSM- GSM - 0x04</li> <li>• DMS_RADIO_IF_UMTS - 0x05</li> <li>• DMS_RADIO_IF_LTE - 0x08</li> <li>• DMS_RADIO_IF_TD - 0x09</li> </ul>

### 8.25.2 Field Documentation

8.25.2.1 uint8\_t dms\_devCaps::DataServiceCapability

8.25.2.2 uint32\_t dms\_devCaps::MaxRXChannelRate

8.25.2.3 uint32\_t dms\_devCaps::MaxTXChannelRate

8.25.2.4 uint8\_t dms\_devCaps::Radiolfaces[255]

8.25.2.5 uint8\_t dms\_devCaps::RadiolfacesSize

8.25.2.6 `uint8_t dms_devCaps::SimCapability`

## 8.26 `dms_devCurSubsCaps` Struct Reference

### Data Fields

- `uint8_t CurSubsCapsLen`
- `uint64_t SubsCapList` [32]

### 8.26.1 Detailed Description

This structure contains the `dms_devCurSubsCaps` response parameters.

#### Parameters

<i>CurSubsCapLen</i>	<ul style="list-style-type: none"> <li>• Length of subscription list</li> </ul>
<i>SubsCapList</i>	<ul style="list-style-type: none"> <li>• Subscription cap list, each array element will be one of following <ul style="list-style-type: none"> <li>– DMS_SUBS_CAPABILITY_AMPS -0x00000001</li> <li>– DMS_SUBS_CAPABILITY_CDMA -0x00000002</li> <li>– DMS_SUBS_CAPABILITY_HDR -0x00000004</li> <li>– DMS_SUBS_CAPABILITY_GSM -0x00000008</li> <li>– DMS_SUBS_CAPABILITY_WCDMA -0x00000010</li> <li>– DMS_SUBS_CAPABILITY_LTE -0x00000020</li> <li>– DMS_SUBS_CAPABILITY_TDS -0x00000040</li> </ul> </li> </ul>

### 8.26.2 Field Documentation

8.26.2.1 `uint8_t dms_devCurSubsCaps::CurSubsCapsLen`

8.26.2.2 `uint64_t dms_devCurSubsCaps::SubsCapList`[32]

## 8.27 `dms_devMaxCfgListCaps` Struct Reference

### Data Fields

- `uint8_t MaxSubs`
- `uint8_t MaxActive`
- `uint8_t DevCfgListLen`
- `dms_devSubsList SubsDevList` [32]
- `uint8_t CurIndex`

### 8.27.1 Detailed Description

This structure contains the `devMaxCfgListCaps` parameters.

## Parameters

<i>MaxSubs</i>	<ul style="list-style-type: none"> <li>Maximum number of subscriptions supported</li> </ul>
<i>MaxActive</i>	<ul style="list-style-type: none"> <li>Maximum number of subscriptions active</li> </ul>
<i>DevCfgListLen</i>	<ul style="list-style-type: none"> <li>Length of subscription config list</li> </ul>
<i>SubsDevList</i>	<ul style="list-style-type: none"> <li>Subscription config list <ul style="list-style-type: none"> <li>See <a href="#">dms_devSubsList</a> for more information</li> </ul> </li> </ul>
<i>CurlIndex</i>	<ul style="list-style-type: none"> <li>Current Config Index</li> </ul>

## 8.27.2 Field Documentation

8.27.2.1 `uint8_t dms_devMaxCfgListCaps::CurlIndex`8.27.2.2 `uint8_t dms_devMaxCfgListCaps::DevCfgListLen`8.27.2.3 `uint8_t dms_devMaxCfgListCaps::MaxActive`8.27.2.4 `uint8_t dms_devMaxCfgListCaps::MaxSubs`8.27.2.5 `dms_devSubsList dms_devMaxCfgListCaps::SubsDevList[32]`

## 8.28 dms\_devMaxSubsCaps Struct Reference

## Data Fields

- `uint8_t` [MaxSubsCapLen](#)
- `uint64_t` [MaxSubsList](#) [32]

## 8.28.1 Detailed Description

This structure contains the [dms\\_devMaxSubsCaps](#) response parameters.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present or not</li> </ul>
<i>MaxSubsCapLen</i>	<ul style="list-style-type: none"> <li>Length of subscription list</li> </ul>

<i>MaxSubsList</i>	<ul style="list-style-type: none"> <li>Subscription cap list, each array element will be one of following <ul style="list-style-type: none"> <li>DMS_SUBS_CAPABILITY_AMPS -0x00000001</li> <li>DMS_SUBS_CAPABILITY_CDMA -0x00000002</li> <li>DMS_SUBS_CAPABILITY_HDR -0x00000004</li> <li>DMS_SUBS_CAPABILITY_GSM -0x00000008</li> <li>DMS_SUBS_CAPABILITY_WCDMA -0x00000010</li> <li>DMS_SUBS_CAPABILITY_LTE -0x00000020</li> <li>DMS_SUBS_CAPABILITY_TDS -0x00000040</li> </ul> </li> </ul>
--------------------	---

## 8.28.2 Field Documentation

8.28.2.1 `uint8_t dms_devMaxSubsCaps::MaxSubsCapLen`

8.28.2.2 `uint64_t dms_devMaxSubsCaps::MaxSubsList[32]`

## 8.29 dms\_devMultiSimCaps Struct Reference

### Data Fields

- `uint8_t` [MaxSubs](#)
- `uint8_t` [SubsCfgListLen](#)
- `dms_devSubsCfgList` [SubsCfgList](#) [32]

### 8.29.1 Detailed Description

This structure contains the [dms\\_devMultiSimCaps](#) parameters.

#### Parameters

<i>MaxSubs</i>	<ul style="list-style-type: none"> <li>Maximum number of subscriptions supported</li> </ul>
<i>SubsCfgListLen</i>	<ul style="list-style-type: none"> <li>Length of subscription config list</li> </ul>
<i>SubsCfgList</i>	<ul style="list-style-type: none"> <li>Subscription config list <ul style="list-style-type: none"> <li>See <code>devSubsCfgList</code> for more information</li> </ul> </li> </ul>

## 8.29.2 Field Documentation

8.29.2.1 `uint8_t dms_devMultiSimCaps::MaxSubs`

8.29.2.2 `dms_devSubsCfgList` `dms_devMultiSimCaps::SubsCfgList`[32]

8.29.2.3 `uint8_t dms_devMultiSimCaps::SubsCfgListLen`



## 8.30 dms\_devMultiSimVoiceDataCaps Struct Reference

### Data Fields

- uint8\_t [MaxSubs](#)
- uint8\_t [MaxActive](#)

### 8.30.1 Detailed Description

This structure contains the [dms\\_devMultiSimVoiceDataCaps](#) parameters.

#### Parameters

<i>MaxSubs</i>	<ul style="list-style-type: none"><li>• Maximum number of subscriptions supported</li></ul>
<i>MaxActive</i>	<ul style="list-style-type: none"><li>• Maximum number of subscriptions active</li></ul>

### 8.30.2 Field Documentation

8.30.2.1 uint8\_t dms\_devMultiSimVoiceDataCaps::MaxActive

8.30.2.2 uint8\_t dms\_devMultiSimVoiceDataCaps::MaxSubs

## 8.31 dms\_devSubsCfgList Struct Reference

### Data Fields

- uint8\_t [MaxActive](#)
- uint8\_t [SubsListLen](#)
- uint64\_t [SubsList](#) [32]

### 8.31.1 Detailed Description

This structure contains the [dms\\_devSubsCfgList](#) parameters.

## Parameters

<i>MaxActive</i>	<ul style="list-style-type: none"> <li>Maximum number of subscriptions active</li> </ul>
<i>SubsListLen</i>	<ul style="list-style-type: none"> <li>Length of subscription list</li> </ul>
<i>SubsList</i>	<ul style="list-style-type: none"> <li>Subscription list, each array element will be one of following <ul style="list-style-type: none"> <li>DMS_SUBS_CAPABILITY_AMPS - 0x00000001</li> <li>DMS_SUBS_CAPABILITY_CDMA - 0x00000002</li> <li>DMS_SUBS_CAPABILITY_HDR - 0x00000004</li> <li>DMS_SUBS_CAPABILITY_GSM - 0x00000008</li> <li>DMS_SUBS_CAPABILITY_WCDMA - 0x00000010</li> <li>DMS_SUBS_CAPABILITY_LTE - 0x00000020</li> <li>DMS_SUBS_CAPABILITY_TDS - 0x00000040</li> </ul> </li> </ul>

## 8.31.2 Field Documentation

8.31.2.1 `uint8_t dms_devSubsCfgList::MaxActive`8.31.2.2 `uint64_t dms_devSubsCfgList::SubsList[32]`8.31.2.3 `uint8_t dms_devSubsCfgList::SubsListLen`8.32 `dms_devSubsFeatureModeCaps` Struct Reference

## Data Fields

- `uint8_t SubsFeatureLen`
- `uint32_t SubsFeatureList [32]`

## 8.32.1 Detailed Description

This structure contains the `dms_devSubsFeatureModeCaps` response parameters.

## Parameters

<i>SubsFeatureLen</i>	<ul style="list-style-type: none"> <li>Length of subscription feature list</li> </ul>
<i>SubsFeatureList</i>	<ul style="list-style-type: none"> <li>Subscription feature mode list, each array element will be one of following <ul style="list-style-type: none"> <li>DMS_DEVICE_SUBS_FEATURE_MODE_NORMAL -0</li> <li>DMS_DEVICE_SUBS_FEATURE_MODE_SGLTE -1</li> <li>DMS_DEVICE_SUBS_FEATURE_MODE_SVLTE -2</li> <li>DMS_DEVICE_SUBS_FEATURE_MODE_SRLTE -3</li> <li>DMS_DEVICE_SUBS_FEATURE_MODE_DUAL_MULTIMODE -4</li> </ul> </li> </ul>

### 8.32.2 Field Documentation

8.32.2.1 `uint8_t dms_devSubsFeatureModeCaps::SubsFeatureLen`

8.32.2.2 `uint32_t dms_devSubsFeatureModeCaps::SubsFeatureList[32]`

## 8.33 dms\_devSubsList Struct Reference

### Data Fields

- `uint8_t SubsListLen`
- `uint64_t SubsList [32]`

### 8.33.1 Detailed Description

This structure contains the [dms\\_devSubsList](#) parameters.

#### Parameters

<i>SubsListLen</i>	<ul style="list-style-type: none"> <li>• Length of subscription list</li> </ul>
<i>SubsList</i>	<ul style="list-style-type: none"> <li>• Subscription list, each array element will be one of following             <ul style="list-style-type: none"> <li>– DMS_SUBS_CAPABILITY_AMPS -0x00000001</li> <li>– DMS_SUBS_CAPABILITY_CDMA -0x00000002</li> <li>– DMS_SUBS_CAPABILITY_HDR -0x00000004</li> <li>– DMS_SUBS_CAPABILITY_GSM -0x00000008</li> <li>– DMS_SUBS_CAPABILITY_WCDMA -0x00000010</li> <li>– DMS_SUBS_CAPABILITY_LTE -0x00000020</li> <li>– DMS_SUBS_CAPABILITY_TDS -0x00000040</li> </ul> </li> </ul>

### 8.33.2 Field Documentation

8.33.2.1 `uint64_t dms_devSubsList::SubsList[32]`

8.33.2.2 `uint8_t dms_devSubsList::SubsListLen`

## 8.34 dms\_devSubsVoiceDataCaps Struct Reference

### Data Fields

- `uint8_t SubsVoiceDataCapLen`
- `dms_devSubsVoiceDataList SubsVoiceDataList [32]`

### 8.34.1 Detailed Description

This structure contains the [dms\\_devSubsVoiceDataCaps](#) response parameters.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>
<i>SubsVoiceData-CapLen</i>	<ul style="list-style-type: none"> <li>• Length of subscription list</li> </ul>
<i>SubsVoiceData-List</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devSubsVoiceDataList</a> for more information</li> </ul>

## 8.34.2 Field Documentation

8.34.2.1 `uint8_t dms_devSubsVoiceDataCaps::SubsVoiceDataCapLen`8.34.2.2 `dms_devSubsVoiceDataList dms_devSubsVoiceDataCaps::SubsVoiceDataList[32]`8.35 `dms_devSubsVoiceDataList` Struct Reference

## Data Fields

- `uint32_t` [SubsVoiceDataCap](#)
- `uint8_t` [SimVoiceDataCap](#)

## 8.35.1 Detailed Description

This structure contains the [dms\\_devSubsVoiceDataList](#) parameters.

## Parameters

<i>SubsVoiceData-Cap</i>	<ul style="list-style-type: none"> <li>• Voice data capabilities of a subscription <ul style="list-style-type: none"> <li>– DMS_SUBS_VOICE_DATA_CAPABILITY_NORMAL -0x01</li> <li>– DMS_SUBS_VOICE_DATA_CAPABILITY_SGLTE - 0x02</li> <li>– DMS_SUBS_VOICE_DATA_CAPABILITY_CSFB -0x03</li> <li>– DMS_SUBS_VOICE_DATA_CAPABILITY_SVLTE -0x04</li> <li>– DMS_SUBS_VOICE_DATA_CAPABILITY_SRLTE -0x05</li> </ul> </li> </ul>
<i>SimVoiceData-Cap</i>	<ul style="list-style-type: none"> <li>• Simultaneous Voice data capabilities of subscription</li> </ul>

## 8.35.2 Field Documentation

8.35.2.1 `uint8_t dms_devSubsVoiceDataList::SimVoiceDataCap`8.35.2.2 `uint32_t dms_devSubsVoiceDataList::SubsVoiceDataCap`8.36 `dms_LteBandsSupport` Struct Reference

## Data Fields

- uint16\_t [supportedLteBandLen](#)
- uint16\_t [lteBands](#) [255]
- uint8\_t [TLVPresent](#)

### 8.36.1 Detailed Description

This structure contains the TLV to get Info of Supported LTE Bands.

#### Parameters

<i>OUT</i>	supportedLteBandLen[OUT] <ul style="list-style-type: none"> <li>• Supported LTE Bands length</li> <li>• Number of set of following elements <ul style="list-style-type: none"> <li>– lteBands</li> </ul> </li> </ul>
<i>OUT</i>	lteBands[OUT] <ul style="list-style-type: none"> <li>• Array of supported LTE bands where each entry is decimal representative of the LTE band supported.</li> </ul>
<i>OUT</i>	TLVPresent[OUT] <ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.36.2 Field Documentation

8.36.2.1 uint16\_t dms\_LteBandsSupport::lteBands[255]

8.36.2.2 uint16\_t dms\_LteBandsSupport::supportedLteBandLen

8.36.2.3 uint8\_t dms\_LteBandsSupport::TLVPresent

## 8.37 dms\_OperatingModeTlv Struct Reference

## Data Fields

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

### 8.37.1 Detailed Description

Operating Mode Tlv

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<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>

## 8.37.2 Field Documentation

8.37.2.1 uint32\_t dms\_OperatingModeTlv::operatingMode

8.37.2.2 uint16\_t dms\_OperatingModeTlv::TlvPresent

## 8.38 dms\_PSMActiveTimerIndTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [ActiveTimerInd](#)

## 8.38.1 Detailed Description

This Structure used to store DMS PSM Active Timer TLV Value.

## 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>ActiveTimerInd</i>	<ul style="list-style-type: none"> <li>• Active Timer</li> <li>• PSM active timer value (in seconds).</li> </ul>

## 8.38.2 Field Documentation

8.38.2.1 uint32\_t dms\_PSMActiveTimerIndTlv::ActiveTimerInd

8.38.2.2 uint8\_t dms\_PSMActiveTimerIndTlv::TlvPresent

## 8.39 dms\_PSMActiveTimerTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [activeTimer](#)

### 8.39.1 Detailed Description

This structure contains the parameters for Power Save Mode active timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>activeTimer</i>	Active Timer <ul style="list-style-type: none"> <li>• Power Save mode active timer value (in seconds).</li> </ul>

### 8.39.2 Field Documentation

8.39.2.1 uint32\_t dms\_PSMActiveTimerTlv::activeTimer

8.39.2.2 uint8\_t dms\_PSMActiveTimerTlv::TlvPresent

## 8.40 dms\_PSMDurationDueToOOSTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [durationDueToOOS](#)

### 8.40.1 Detailed Description

This structure contains the parameters for Power Save Mode Duration due to an outage.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>durationDueToOOS</i>	Duration Due to OOS <ul style="list-style-type: none"> <li>• Power Save mode duration due to an outage (in seconds).</li> </ul>

### 8.40.2 Field Documentation

8.40.2.1 uint32\_t dms\_PSMDurationDueToOOSTlv::durationDueToOOS

8.40.2.2 uint8\_t dms\_PSMDurationDueToOOSTlv::TlvPresent

## 8.41 dms\_PSMDurationThresholdTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [durationThreshold](#)

### 8.41.1 Detailed Description

This structure contains the parameters for Power Save Mode Duration Threshold.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>duration-Threshold</i>	Duration Threshold <ul style="list-style-type: none"> <li>• Minimum duration for the device to benefit by entering PSM (in seconds).</li> </ul>

### 8.41.2 Field Documentation

8.41.2.1 uint32\_t dms\_PSMDurationThresholdTlv::durationThreshold

8.41.2.2 uint8\_t dms\_PSMDurationThresholdTlv::TlvPresent

## 8.42 dms\_PSMEarlyWakeupTimeTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [earlyWakeupTime](#)

### 8.42.1 Detailed Description

This structure contains the parameters for Power Save Mode early wakeup time.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>earlyWakeup-Time</i>	<ul style="list-style-type: none"> <li>• Early Wakeup Time</li> <li>• Power Save mode early wakeup time (in seconds) indicating how early the device should exit PSM to offset for bootup and acquisition delay.</li> </ul>

### 8.42.2 Field Documentation

8.42.2.1 uint32\_t dms\_PSMEarlyWakeupTimeTlv::earlyWakeupTime

8.42.2.2 uint8\_t dms\_PSMEarlyWakeupTimeTlv::TlvPresent



## 8.43 dms\_PSMEnableStateIndTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [EnableStateInd](#)

#### 8.43.1 Detailed Description

This Structure used to store DMS PSM Enable state TLV Value.

##### 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>EnableStateInd</i>	<ul style="list-style-type: none"> <li>• PSM Enable State</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - PSM is not enabled</li> <li>– 1 - PSM is enabled</li> </ul> </li> </ul>

#### 8.43.2 Field Documentation

8.43.2.1 uint8\_t dms\_PSMEnableStateIndTlv::EnableStateInd

8.43.2.2 uint8\_t dms\_PSMEnableStateIndTlv::TlvPresent

## 8.44 dms\_PSMEnableStateTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [enableState](#)

#### 8.44.1 Detailed Description

This structure contains the parameters for Power Save Mode Enable State.

##### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>enableState</i>	Power Save Mode Enable State Values <ul style="list-style-type: none"> <li>• 0 - PSM is not enabled</li> <li>• 1 - PSM is enabled</li> </ul>

#### 8.44.2 Field Documentation

8.44.2.1 `uint8_t dms_PSMEnableStateTlv::enableState`

8.44.2.2 `uint8_t dms_PSMEnableStateTlv::TlvPresent`

## 8.45 dms\_PSMPeriodicUpdateTimerIndTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t PeriodicUpdateTimerInd`

### 8.45.1 Detailed Description

This Structure used to store DMS PSM Periodic Update Timer TLV Value.

#### 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>PeriodicUpdate-TimerInd</i>	<ul style="list-style-type: none"> <li>• Periodic Update Timer</li> <li>• PSM periodic update timer value (in seconds).</li> </ul>

### 8.45.2 Field Documentation

8.45.2.1 `uint32_t dms_PSMPeriodicUpdateTimerIndTlv::PeriodicUpdateTimerInd`

8.45.2.2 `uint8_t dms_PSMPeriodicUpdateTimerIndTlv::TlvPresent`

## 8.46 dms\_PSMPeriodicUpdateTimerTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t periodicUpdateTimer`

### 8.46.1 Detailed Description

This structure contains the parameters for Power Save Mode periodic update timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>periodicUpdate-Timer</i>	Periodic Update Timer <ul style="list-style-type: none"> <li>• Power Save mode periodic update timer value (in seconds).</li> </ul>

## 8.46.2 Field Documentation

8.46.2.1 uint32\_t dms\_PSMPeriodicUpdateTimerTlv::periodicUpdateTimer

8.46.2.2 uint8\_t dms\_PSMPeriodicUpdateTimerTlv::TlvPresent

## 8.47 dms\_PSMRandomizationWindowTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [randomizationWindow](#)

### 8.47.1 Detailed Description

This structure contains the parameters for Power Save Mode wakeup randomization window.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>randomization-Window</i>	Randomization Window <ul style="list-style-type: none"> <li>• Power Save mode wakeup randomization window (in seconds)</li> </ul>

## 8.47.2 Field Documentation

8.47.2.1 uint32\_t dms\_PSMRandomizationWindowTlv::randomizationWindow

8.47.2.2 uint8\_t dms\_PSMRandomizationWindowTlv::TlvPresent

## 8.48 dms\_TemperatureTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [TempStat](#)
- uint16\_t [Temperature](#)

### 8.48.1 Detailed Description

This structure stores information about temperature TLV.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>TempStat</i>	<ul style="list-style-type: none"> <li>• Temperature state <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>Temperature</i>	<ul style="list-style-type: none"> <li>• temperature in degree celsius</li> </ul>

## 8.48.2 Field Documentation

8.48.2.1 uint16\_t dms\_TemperatureTlv::Temperature

8.48.2.2 uint8\_t dms\_TemperatureTlv::TempStat

8.48.2.3 uint8\_t dms\_TemperatureTlv::TlvPresent

## 8.49 dms\_UimAutoSwitchActSlotTlv Struct Reference

## Data Fields

- uint8\_t [uimAutoSwitchActSlot](#)
- uint8\_t [TlvPresent](#)

## 8.49.1 Detailed Description

This structure contains the TLV required to Get UIM auto active slot Tlv.

## Parameters

<i>uimAutoSwitch-ActSlot[Optional]</i>	<ul style="list-style-type: none"> <li>• When UIMAUTOSWITCH feature enabled and SIM auto-switch is activated, indicates currently active UIM slot.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - slot 1 (e.g. external SIM)</li> <li>– 1 - slot 2 (e.g. embedded SIM)</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.49.2 Field Documentation

8.49.2.1 uint8\_t dms\_UimAutoSwitchActSlotTlv::TlvPresent

8.49.2.2 `uint8_t dms_UimAutoSwitchActSlotTlv::uimAutoSwitchActSlot`

## 8.50 dms\_UimStatusTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t intf`
- `uint8_t event`

### 8.50.1 Detailed Description

This structure stores information about UIM status TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>intf</i>	<ul style="list-style-type: none"> <li>• interface type <ul style="list-style-type: none"> <li>– 0 - External UIM.</li> <li>– 1 - Embedded UIM.</li> <li>– 2 - Remote UIM.</li> </ul> </li> </ul>
<i>event</i>	<ul style="list-style-type: none"> <li>• event type <ul style="list-style-type: none"> <li>– 0 - UIM card removed.</li> <li>– 1 - UIM card inserted.</li> </ul> </li> </ul>

### 8.50.2 Field Documentation

8.50.2.1 `uint8_t dms_UimStatusTlv::event`

8.50.2.2 `uint8_t dms_UimStatusTlv::intf`

8.50.2.3 `uint8_t dms_UimStatusTlv::TlvPresent`

## 8.51 dms\_VoltageTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t VoltStat`
- `uint16_t Voltage`

### 8.51.1 Detailed Description

This structure stores information about voltage TLV.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>VoltStat</i>	<ul style="list-style-type: none"> <li>• Voltage state <ul style="list-style-type: none"> <li>– 0 - unknown</li> <li>– 1 - normal</li> <li>– 2 - low (warning)</li> <li>– 3 - low (critical)</li> <li>– 4 - high (critical)</li> </ul> </li> </ul>
<i>Voltage</i>	<ul style="list-style-type: none"> <li>• Voltage in mV</li> </ul>

## 8.51.2 Field Documentation

8.51.2.1 uint8\_t dms\_VoltageTlv::TlvPresent

8.51.2.2 uint16\_t dms\_VoltageTlv::Voltage

8.51.2.3 uint8\_t dms\_VoltageTlv::VoltStat

## 8.52 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](#)

## 8.52.1 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 <a href="#">cust_id</a> field</li> </ul>
<i>cust_id</i>	<ul style="list-style-type: none"> <li>• Customization ID (Maximum 64 bytes)</li> <li>• NULL terminated ASCII string.</li> </ul>
<i>value_length</i>	<ul style="list-style-type: none"> <li>• length of <a href="#">cust_value</a> 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>
------------------	--

## 8.52.2 Field Documentation

8.52.2.1 `uint16_t DMScustSettingInfo::cust_attr`

8.52.2.2 `uint8_t DMScustSettingInfo::cust_id[64+1]`

8.52.2.3 `uint8_t DMScustSettingInfo::cust_value[8+1]`

8.52.2.4 `uint16_t DMScustSettingInfo::id_length`

8.52.2.5 `uint16_t DMScustSettingInfo::value_length`

## 8.53 DMScustSettingList Struct Reference

### Data Fields

- `uint8_t list_type`
- `uint16_t num_instances`
- `DMScustSettingInfo custSetting [255+1]`

### 8.53.1 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="#">DMScustSettingInfo</a> for more information</li> </ul>

## 8.53.2 Field Documentation

8.53.2.1 `DMScustSettingInfo DMScustSettingList::custSetting[255+1]`

8.53.2.2 `uint8_t DMScustSettingList::list_type`

8.53.2.3 `uint16_t DMScustSettingList::num_instances`

## 8.54 DMSgetCustomFeatureV2 Struct Reference

### Data Fields

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

### 8.54.1 Detailed Description

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

#### Parameters

<i>pGetCustom-Input[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">DMSgetCustomInput</a> for more information</li> </ul>
<i>pCustSetting-Info[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">DMScustSettingInfo</a> for more information</li> </ul>
<i>pCustSetting-List[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">DMScustSettingList</a> for more information</li> </ul>

### 8.54.2 Field Documentation

8.54.2.1 [DMScustSettingInfo](#)\* [DMSgetCustomFeatureV2::pCustSettingInfo](#)

8.54.2.2 [DMScustSettingList](#)\* [DMSgetCustomFeatureV2::pCustSettingList](#)

8.54.2.3 [DMSgetCustomInput](#)\* [DMSgetCustomFeatureV2::pGetCustomInput](#)

## 8.55 DMSgetCustomInput Struct Reference

### Data Fields

- `uint8_t` [cust\\_id](#) [64+1]
- `uint8_t` [list\\_type](#)

### 8.55.1 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> <li>• NULL terminated ASCII string.</li> </ul>
<i>list_type</i>	<ul style="list-style-type: none"> <li>• list type requested</li> </ul>



### 8.55.2 Field Documentation

8.55.2.1 `uint8_t DMSgetCustomInput::cust_id[64+1]`

8.55.2.2 `uint8_t DMSgetCustomInput::list_type`

## 8.56 dunchannelRate Struct Reference

### Data Fields

- `uint32_t CurrChanTxRate`
- `uint32_t CurrChanRxRate`
- `uint32_t MaxChanTxRate`
- `uint32_t MaxChanRxRate`

### 8.56.1 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> <li>• 0xffffffff - Invalid data.</li> </ul>
<i>CurrChanRxRate</i>	<ul style="list-style-type: none"> <li>• Instantaneous channel Rx rate in bits per second</li> <li>• 0xffffffff - Invalid data.</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> <li>• 0xffffffff - Invalid data.</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> <li>• 0xffffffff - Invalid data.</li> </ul>

### 8.56.2 Field Documentation

8.56.2.1 `uint32_t dunchannelRate::CurrChanRxRate`

8.56.2.2 `uint32_t dunchannelRate::CurrChanTxRate`

8.56.2.3 `uint32_t dunchannelRate::MaxChanRxRate`

8.56.2.4 `uint32_t dunchannelRate::MaxChanTxRate`

## 8.57 eriDataparams Struct Reference

### Data Fields

- `uint16_t eriDataLen`

- `uint8_t eriData` [1024]

### 8.57.1 Detailed Description

This structure contains Extended Roaming Indicator(ERI) file parameters

#### Parameters

<i>eriDataLen</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>eriData</i>	<ul style="list-style-type: none"> <li>• ERI data read from persistent storage( Max size is 1024 )</li> </ul>

### 8.57.2 Field Documentation

8.57.2.1 `uint8_t eriDataparams::eriData`[1024]

8.57.2.2 `uint16_t eriDataparams::eriDataLen`

## 8.58 eTWSPLMNInfoTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `sMSEtwsPlmnInfo` `ETWSPLMNInfo`

### 8.58.1 Detailed Description

This structure contains ETWS PLMN TLV information

#### 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 <code>sMSEtwsPlmnInfo</code> for more information</li> </ul>

### 8.58.2 Field Documentation

8.58.2.1 `sMSEtwsPlmnInfo` `eTWSPLMNInfoTlv::ETWSPLMNInfo`

8.58.2.2 `uint8_t eTWSPLMNInfoTlv::TlvPresent`

## 8.59 FMSImageElement Struct Reference

## Data Fields

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

### 8.59.1 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>

### 8.59.2 Field Documentation

8.59.2.1 [uint8\\_t FMSImageElement::buildId](#)[100]

8.59.2.2 [uint8\\_t FMSImageElement::buildIdLength](#)

8.59.2.3 [uint8\\_t FMSImageElement::imageId](#)[100]

8.59.2.4 [uint8\\_t FMSImageElement::imageType](#)

## 8.60 FMSImageIdElement Struct Reference

## Data Fields

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

### 8.60.1 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>

## 8.60.2 Field Documentation

8.60.2.1 `uint8_t FMSImageIDElement::buildID[100]`8.60.2.2 `uint8_t FMSImageIDElement::buildIDLength`8.60.2.3 `uint8_t FMSImageIDElement::failureCount`8.60.2.4 `uint8_t FMSImageIDElement::imageID[100]`8.60.2.5 `uint8_t FMSImageIDElement::storageIndex`

## 8.61 FMSImageIDEntries Struct Reference

## Data Fields

- `uint8_t imageType`
- `uint8_t maxImages`
- `uint8_t executingImage`
- `uint8_t imageIDSize`
- `FMSImageIDElement imageIDElement [50]`

## 8.61.1 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>

## 8.61.2 Field Documentation

8.61.2.1 `uint8_t FMSImageIDEntries::executingImage`

8.61.2.2 `FMSImageIDElement FMSImageIDEntries::imageIDElement[50]`

8.61.2.3 `uint8_t FMSImageIDEntries::imageIDSize`

8.61.2.4 `uint8_t FMSImageIDEntries::imageType`

8.61.2.5 `uint8_t FMSImageIDEntries::maxImages`

## 8.62 FMSImageList Struct Reference

### Data Fields

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

### 8.62.1 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 ImageIDEntries Structure ( Max 2 entries )</li> </ul>

## 8.62.2 Field Documentation

8.62.2.1 `FMSImageIDEntries FMSImageList::imageIDEntries[2]`

8.62.2.2 `uint8_t FMSImageList::listSize`

## 8.63 FMSPrefImageList Struct Reference

### Data Fields

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

### 8.63.1 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>

### 8.63.2 Field Documentation

#### 8.63.2.1 FMSImageElement FMSPrefImageList::listEntries[2]

#### 8.63.2.2 uint8\_t FMSPrefImageList::listSize

## 8.64 hdrSSInfo Struct Reference

### Data Fields

- [int8\\_t rssi](#)
- [int16\\_t ecio](#)
- [uint8\\_t sinr](#)
- [int32\\_t io](#)

### 8.64.1 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>

## 8.64.2 Field Documentation

8.64.2.1 `int16_t` `hdrSSInfo::ecio`

8.64.2.2 `int32_t` `hdrSSInfo::io`

8.64.2.3 `int8_t` `hdrSSInfo::rssi`

8.64.2.4 `uint8_t` `hdrSSInfo::sinr`

## 8.65 image\_info\_t Struct Reference

### Data Fields

- `uint8_t` [imageType](#)
- `uint8_t` [uniqueID](#) [16]
- `uint8_t` [buildIDLen](#)
- `uint8_t` [buildID](#) [255]

### 8.65.1 Detailed Description

This structure is used to store image information

#### Parameters

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

## 8.65.2 Field Documentation

8.65.2.1 `uint8_t image_info_t::buildID[255]`

8.65.2.2 `uint8_t image_info_t::buildIDLen`

8.65.2.3 `uint8_t image_info_t::imageType`

8.65.2.4 `uint8_t image_info_t::uniqueID[16]`

## 8.66 ims\_AMRModelInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint8_t` [amrMode](#)

### 8.66.1 Detailed Description

This structure store information about AMR NB mode.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>amrMode</i>	Bitmask indicating AMR modes. 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>

## 8.66.2 Field Documentation



8.66.2.1 `uint8_t` `ims_AMRModelInfo::amrMode`

8.66.2.2 `uint8_t` `ims_AMRModelInfo::TlvPresent`

## 8.67 ims\_AMROctAlgnInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint8_t` [amrOctAlgn](#)

### 8.67.1 Detailed Description

This structure store information about AMR NB octet aligned.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>amrOctAlgn</i>	-1- if octet aligned -0- if octed not aligned

### 8.67.2 Field Documentation

8.67.2.1 `uint8_t` `ims_AMROctAlgnInfo::amrOctAlgn`

8.67.2.2 `uint8_t` `ims_AMROctAlgnInfo::TlvPresent`

## 8.68 ims\_AMRWBModelInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint16_t` [amrWBMode](#)

### 8.68.1 Detailed Description

This structure store information about AMR WB mode.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>amrWBMode</i>	Bitmask indicating AMR WB modes. 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>

## 8.68.2 Field Documentation

8.68.2.1 uint16\_t ims\_AMRWBModelInfo::amrWBMode

8.68.2.2 uint8\_t ims\_AMRWBModelInfo::TlvPresent

## 8.69 ims\_AMRWBOctAlignInfo Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [amrWBOctAlign](#)

## 8.69.1 Detailed Description

This structure store information about AMR WB octet aligned.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>amrWBOctAlign</i>	-1- if octet aligned -0- if octed not aligned

## 8.69.2 Field Documentation

8.69.2.1 uint8\_t ims\_AMRWBOctAlignInfo::amrWBOctAlign

8.69.2.2 uint8\_t ims\_AMRWBOctAlignInfo::TlvPresent

## 8.70 ims\_CSCFPortNameInfo Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [cscfPortName](#) [255]

### 8.70.1 Detailed Description

This structure hold parameters about CSCF port info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>cscfPortName</i>	-CSCF port name string

### 8.70.2 Field Documentation

8.70.2.1 `uint8_t` `ims_CSCFPortNameInfo::cscfPortName[255]`

8.70.2.2 `uint8_t` `ims_CSCFPortNameInfo::TlvPresent`

## 8.71 ims\_EnabAMRWBInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint8_t` [amrWBEnable](#)

### 8.71.1 Detailed Description

This structure store information about Enable AMR WB.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>amrWBEnable</i>	-1- Enable -0- Disable

### 8.71.2 Field Documentation

8.71.2.1 `uint8_t` `ims_EnabAMRWBInfo::amrWBEnable`

8.71.2.2 `uint8_t` `ims_EnabAMRWBInfo::TlvPresent`

## 8.72 ims\_EnabSCRAMRInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint8_t` [scrAmrEnable](#)

### 8.72.1 Detailed Description

This structure store information about Enable SCR AMR .

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>scrAmrEnable</i>	-1- Enable -0- Disable

## 8.72.2 Field Documentation

8.72.2.1 `uint8_t ims_EnabSCRAMRInfo::scrAmrEnable`8.72.2.2 `uint8_t ims_EnabSCRAMRInfo::TlvPresent`8.73 `ims_EnabSCRAMRWBInfo` Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `uint8_t scrAmrWBEnable`

## 8.73.1 Detailed Description

This structure store information about Enable SCR AMR WB.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>scrAmrWB-Enable</i>	-1- Enable -0- Disable

## 8.73.2 Field Documentation

8.73.2.1 `uint8_t ims_EnabSCRAMRWBInfo::scrAmrWBEnable`8.73.2.2 `uint8_t ims_EnabSCRAMRWBInfo::TlvPresent`8.74 `ims_IMSDomainInfo` Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `uint8_t imsDomainName` [255]

## 8.74.1 Detailed Description

This structure hold parameters about IMS domain info.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
-------------------	--

<i>imsDomainName</i>	-IMS domain name string
----------------------	-------------------------

### 8.74.2 Field Documentation

8.74.2.1 `uint8_t` `ims_IMSDomainInfo::imsDomainName[255]`

8.74.2.2 `uint8_t` `ims_IMSDomainInfo::TlvPresent`

## 8.75 ims\_IMSTestModelInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint8_t` [imsTestMode](#)

### 8.75.1 Detailed Description

This structure hold parameters about ims test mode info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>imsTestMode</i>	-1- if IMS test mode is enabled -0- if IMS test mode is disabled

### 8.75.2 Field Documentation

8.75.2.1 `uint8_t` `ims_IMSTestModelInfo::imsTestMode`

8.75.2.2 `uint8_t` `ims_IMSTestModelInfo::TlvPresent`

## 8.76 ims\_MinSessExpInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint16_t` [minSessExp](#)

### 8.76.1 Detailed Description

This structure store information about minimum session expiry info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>minSessExp</i>	<ul style="list-style-type: none"><li>• minimum session expiry in seconds</li></ul>

## 8.76.2 Field Documentation

8.76.2.1 uint16\_t `ims_MinSessExpInfo::minSessExp`

8.76.2.2 uint8\_t `ims_MinSessExpInfo::TlvPresent`

## 8.77 `ims_PCSCFPortInfo` Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [priCSCFPort](#)

### 8.77.1 Detailed Description

This structure hold parameters about primary CSCF port info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>priCSCFPort</i>	<ul style="list-style-type: none"> <li>• primary CSCF port number</li> </ul>

## 8.77.2 Field Documentation

8.77.2.1 uint16\_t `ims_PCSCFPortInfo::priCSCFPort`

8.77.2.2 uint8\_t `ims_PCSCFPortInfo::TlvPresent`

## 8.78 `ims_PhCtxtURIInfo` Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [PhCtxtURI](#) [255]

### 8.78.1 Detailed Description

This structure hold parameters about Phone context URI info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>PhCtxtURI</i>	-Phone context URI string

## 8.78.2 Field Documentation

8.78.2.1 uint8\_t ims\_PhCtxtURIInfo::PhCtxtURI[255]

8.78.2.2 uint8\_t ims\_PhCtxtURIInfo::TlvPresent

## 8.79 ims\_RngBkTmrInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [RingBkTmr](#)

### 8.79.1 Detailed Description

This structure hold parameters about Ring back timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>RingBkTmr</i>	<ul style="list-style-type: none"><li>• Duration, in seconds, of the Ringback timer</li></ul>

### 8.79.2 Field Documentation

8.79.2.1 uint16\_t ims\_RngBkTmrInfo::RingBkTmr

8.79.2.2 uint8\_t ims\_RngBkTmrInfo::TlvPresent

## 8.80 ims\_RngTmrInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [RingTmr](#)

### 8.80.1 Detailed Description

This structure hold parameters about Ring timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>RingTmr</i>	<ul style="list-style-type: none"><li>• Duration, in seconds, of the Ring timer</li></ul>

### 8.80.2 Field Documentation

8.80.2.1 uint16\_t `ims_RngTmrInfo::RingTmr`

8.80.2.2 uint8\_t `ims_RngTmrInfo::TlvPresent`

## 8.81 `ims_RTPRTCPInactTmrDurInfo` Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [InactTmr](#)

#### 8.81.1 Detailed Description

This structure hold parameters about RTP/RTCP timer.

##### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>InactTmr</i>	<ul style="list-style-type: none"> <li>• Duration, in seconds, of the RTP/RTCP inactivity timer</li> </ul>

#### 8.81.2 Field Documentation

8.81.2.1 uint16\_t `ims_RTPRTCPInactTmrDurInfo::InactTmr`

8.81.2.2 uint8\_t `ims_RTPRTCPInactTmrDurInfo::TlvPresent`

## 8.82 `ims_SessDurInfo` Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [sessExp](#)

#### 8.82.1 Detailed Description

This structure store information about session duration info.

##### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>sessExp</i>	<ul style="list-style-type: none"> <li>• session duration in seconds</li> </ul>

#### 8.82.2 Field Documentation



8.82.2.1 uint16\_t ims\_SessDurInfo::sessExp

8.82.2.2 uint8\_t ims\_SessDurInfo::TlvPresent

## 8.83 ims\_SigCompEnInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [SigCompEn](#)

### 8.83.1 Detailed Description

This structure hold parameters about SigComp status.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>SigCompEn</i>	-1- if SigComp is enabled -0- if IMS SigComp is disabled

### 8.83.2 Field Documentation

8.83.2.1 uint8\_t ims\_SigCompEnInfo::SigCompEn

8.83.2.2 uint8\_t ims\_SigCompEnInfo::TlvPresent

## 8.84 ims\_SIPPortInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [SIPLocalPort](#)

### 8.84.1 Detailed Description

This structure stores information about SIP port info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>SIPLocalPort</i>	-SIP local port

### 8.84.2 Field Documentation

8.84.2.1 uint16\_t ims\_SIPPortInfo::SIPLocalPort

8.84.2.2 uint8\_t ims\_SIPPortInfo::TlvPresent

## 8.85 ims\_SIPRegnTmrInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [tmrSIPRegn](#)

### 8.85.1 Detailed Description

This structure stores information about SIP registration timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>tmrSIPRegn</i>	-Initial SIP registration duration, in seconds

### 8.85.2 Field Documentation

8.85.2.1 uint8\_t `ims_SIPRegnTmrInfo::TlvPresent`

8.85.2.2 uint32\_t `ims_SIPRegnTmrInfo::tmrSIPRegn`

## 8.86 ims\_SMSFmtInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [smsFormat](#)

### 8.86.1 Detailed Description

This structure hold parameters about SMS format info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>smsFormat</i>	<ul style="list-style-type: none"> <li>• 0- IMS_SETTINGS_SMS_FORMAT_3GPP2</li> <li>• 1- IMS_SETTINGS_SMS_FORMAT_3GPP</li> </ul>

### 8.86.2 Field Documentation

8.86.2.1 uint8\_t `ims_SMSFmtInfo::smsFormat`

8.86.2.2 uint8\_t `ims_SMSFmtInfo::TlvPresent`

## 8.87 ims\_SMSolPNwInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [smsolPNW](#)

### 8.87.1 Detailed Description

This structure hold parameters about SMS over IP network.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>smsolPNW</i>	-1- if MO SMS turned on -0- if MO SMS turned off

### 8.87.2 Field Documentation

8.87.2.1 uint8\_t `ims_SMSolPNwInfo::smsolPNW`

8.87.2.2 uint8\_t `ims_SMSolPNwInfo::TlvPresent`

## 8.88 ims\_SubscrTmrInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [subscrTmr](#)

### 8.88.1 Detailed Description

This structure stores information about subscriber timer.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>subscrTmr</i>	-Duration, in seconds, of the subscription by the UE for IMS registration notifications.

### 8.88.2 Field Documentation

8.88.2.1 uint32\_t `ims_SubscrTmrInfo::subscrTmr`

8.88.2.2 uint8\_t `ims_SubscrTmrInfo::TlvPresent`

## 8.89 ims\_TmrT1Info Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [tmrT1](#)

### 8.89.1 Detailed Description

This structure stores information about timer T1.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>tmrT1</i>	-RTT estimate, in milliseconds.

### 8.89.2 Field Documentation

8.89.2.1 uint8\_t `ims_TmrT1Info::TlvPresent`

8.89.2.2 uint32\_t `ims_TmrT1Info::tmrT1`

## 8.90 ims\_TmrT2Info Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [tmrT2](#)

### 8.90.1 Detailed Description

This structure stores information about timer T2.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>tmrT2</i>	-Maximum retransmit interval, in msec, for non-invite requests and invite responses.

### 8.90.2 Field Documentation

8.90.2.1 uint8\_t `ims_TmrT2Info::TlvPresent`

8.90.2.2 uint32\_t `ims_TmrT2Info::tmrT2`

## 8.91 ims\_TmrTfInfo Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [tmrTf](#)

### 8.91.1 Detailed Description

This structure store information about timer TF.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>tmrTf</i>	-Non-invite transaction timeout timer, in msec

### 8.91.2 Field Documentation

8.91.2.1 `uint8_t` `ims_TmrTfInfo::TlvPresent`

8.91.2.2 `uint32_t` `ims_TmrTfInfo::tmrTf`

## 8.92 imsa\_IMSFailErrCodeTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint32_t` [ImsFailErrCode](#)

### 8.92.1 Detailed Description

This structure hold parameters about IMS failure code for PDP connection.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>ImsFailErrCode</i>	<ul style="list-style-type: none"> <li>0 -IMSA_PDP_STATUS_OTHER_FAILURE - Generic failure reason</li> <li>1 -IMSA_PDP_STATUS_OPTION_UNSUBSCRIBED - Option is unsubscribed</li> <li>2 -IMSA_PDP_STATUS_UNKNOWN_PDP - PDP was unknown</li> </ul>

### 8.92.2 Field Documentation

8.92.2.1 `uint32_t` `imsa_IMSFailErrCodeTlv::ImsFailErrCode`

8.92.2.2 `uint8_t` `imsa_IMSFailErrCodeTlv::TlvPresent`

## 8.93 imsa\_IMSRegStatusErrorCodeInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint16_t` [ErrorCode](#)

### 8.93.1 Detailed Description

This structure hold parameters about IMS registration error info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>ErrorCode</i>	Error codes when registration status is IMSA_STATUS_NOT_REGISTERED <ul style="list-style-type: none"> <li>• Values:</li> <li>• 3xx - Redirection responses</li> <li>• 4xx - Client failure responses</li> <li>• 5xx - Server failure responses</li> <li>• 6xx - Global failure responses</li> </ul>

### 8.93.2 Field Documentation

8.93.2.1 `uint16_t imsa_IMSRegStatusErrorCodeInfo::ErrorCode`

8.93.2.2 `uint8_t imsa_IMSRegStatusErrorCodeInfo::TlvPresent`

## 8.94 imsa\_IMSRegStatusInfo Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t ImsRegistered`

### 8.94.1 Detailed Description

This structure hold parameters about IMS registration info (deprecated tlv).

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>ImsRegistered</i>	<ul style="list-style-type: none"> <li>• Values:</li> <li>• 0 -Not registered</li> <li>• 1- Registered</li> </ul>

### 8.94.2 Field Documentation

8.94.2.1 `uint8_t imsa_IMSRegStatusInfo::ImsRegistered`

8.94.2.2 `uint8_t imsa_IMSRegStatusInfo::TlvPresent`

## 8.95 imsa\_NewIMSRegStatusInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [ImsRegStatus](#)

### 8.95.1 Detailed Description

This structure hold parameters about IMS registration info (new tlv).

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>ImsRegStatus</i>	<ul style="list-style-type: none"> <li>• Values:</li> <li>• 0 -IMSA_STATUS_NOT_REGISTERED- Not registered for IMS</li> <li>• 1- IMSA_STATUS_REGISTERING - Registering for IMS</li> <li>• 2- IMSA_STATUS_REGISTERED - Registered for IMS</li> </ul>

### 8.95.2 Field Documentation

8.95.2.1 uint32\_t imsa\_NewIMSRegStatusInfo::ImsRegStatus

8.95.2.2 uint8\_t imsa\_NewIMSRegStatusInfo::TlvPresent

## 8.96 imsa\_RatHandoverStatusInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [RatHandoverStatus](#)
- uint32\_t [SourceRAT](#)
- uint32\_t [TargetRAT](#)
- uint8\_t [ErrorCodeLen](#)
- uint8\_t [ErrorCodeData](#) [256]

### 8.96.1 Detailed Description

This structure hold parameters about RAT handover status.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>RatHandover-Status</i>	<ul style="list-style-type: none"> <li>• RAT handover status</li> </ul>

<i>SourceRAT</i>	<ul style="list-style-type: none"> <li>source RAT info: IWLAN/WWAN</li> </ul>
<i>TargetRAT</i>	<ul style="list-style-type: none"> <li>target TAT info: IWLAN/WWAN</li> </ul>
<i>ErrorCodeLen</i>	<ul style="list-style-type: none"> <li>error code length</li> </ul>
<i>ErrorCodeData</i>	<ul style="list-style-type: none"> <li>handover failure code string when status is IMSA_STATUS_RAT_HO_FAILURE</li> </ul>

## 8.96.2 Field Documentation

8.96.2.1 uint8\_t imsa\_RatHandoverStatusInfo::ErrorCodeData[256]

8.96.2.2 uint8\_t imsa\_RatHandoverStatusInfo::ErrorCodeLen

8.96.2.3 uint32\_t imsa\_RatHandoverStatusInfo::RatHandoverStatus

8.96.2.4 uint32\_t imsa\_RatHandoverStatusInfo::SourceRAT

8.96.2.5 uint32\_t imsa\_RatHandoverStatusInfo::TargetRAT

8.96.2.6 uint8\_t imsa\_RatHandoverStatusInfo::TlvPresent

## 8.97 imsa\_SmsRatInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [SmsRatVal](#)

### 8.97.1 Detailed Description

This structure hold parameters about SMS RAT info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>SmsRatVal</i>	<ul style="list-style-type: none"> <li>Values:</li> <li>0-IMSA_WLAN - IMS service is registered on WLAN</li> <li>1-IMSA_WWAN - IMS service is registered on WWAN</li> <li>2-IMSA_IWLAN - IMS service is registered on interworking WLAN</li> </ul>

## 8.97.2 Field Documentation



8.97.2.1 uint32\_t imsa\_SmsRatInfo::SmsRatVal

8.97.2.2 uint8\_t imsa\_SmsRatInfo::TlvPresent

## 8.98 imsa\_SmsSvcStatusInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [SmsSvcStatus](#)

#### 8.98.1 Detailed Description

This structure hold parameters about SMS service info.

##### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>SmsSvcStatus</i>	<ul style="list-style-type: none"><li>Values: -0 - IMS SMS service is not available -1 - IMS SMS is in limited service -2 - IMS SMS is in full service</li></ul>

#### 8.98.2 Field Documentation

8.98.2.1 uint32\_t imsa\_SmsSvcStatusInfo::SmsSvcStatus

8.98.2.2 uint8\_t imsa\_SmsSvcStatusInfo::TlvPresent

## 8.99 imsa\_UtRatInfo Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [UtRatVal](#)

#### 8.99.1 Detailed Description

This structure hold parameters about UT RAT info.

##### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>UtRatVal</i>	<ul style="list-style-type: none"><li>Values:<ul style="list-style-type: none"><li>0-IMSA_WLAN - IMS service is registered on WLAN</li><li>1-IMSA_WWAN - IMS service is registered on WWAN</li><li>2-IMSA_IWLAN - IMS service is registered on interworking WLAN</li></ul></li></ul>

## 8.99.2 Field Documentation

8.99.2.1 `uint8_t imsa_UtRatInfo::TlvPresent`

8.99.2.2 `uint32_t imsa_UtRatInfo::UtRatVal`

## 8.100 imsa\_UtSvcStatusInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint32_t` [UtSvcStatus](#)

### 8.100.1 Detailed Description

This structure hold parameters about UT service info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>UtSvcStatus</i>	<ul style="list-style-type: none"> <li>• Values: -0 - IMS UT service is not available -2 - IMS UT is in full service</li> </ul>

## 8.100.2 Field Documentation

8.100.2.1 `uint8_t imsa_UtSvcStatusInfo::TlvPresent`

8.100.2.2 `uint32_t imsa_UtSvcStatusInfo::UtSvcStatus`

## 8.101 imsa\_VoipRatInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint32_t` [VoipRatVal](#)

### 8.101.1 Detailed Description

This structure hold parameters about VOIP RAT info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>VoipRatVal</i>	<ul style="list-style-type: none"> <li>• Values:</li> <li>• 0-IMSA_WLAN - IMS service is registered on WLAN</li> <li>• 1-IMSA_WWAN - IMS service is registered on WWAN</li> <li>• 2-IMSA_IWLAN - IMS service is registered on interworking WLAN</li> </ul>

### 8.101.2 Field Documentation

8.101.2.1 `uint8_t imsa_VoipRatInfo::TlvPresent`

8.101.2.2 `uint32_t imsa_VoipRatInfo::VoipRatVal`

## 8.102 imsa\_VoipSvcStatusInfo Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t VoipSvcStatus`

### 8.102.1 Detailed Description

This structure hold parameters about VOIP service info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>VoipSvcStatus</i>	<ul style="list-style-type: none"><li>• Values: -0 - IMS VOIP service is not available -2 - IMS VOIP is in full service</li></ul>

### 8.102.2 Field Documentation

8.102.2.1 `uint8_t imsa_VoipSvcStatusInfo::TlvPresent`

8.102.2.2 `uint32_t imsa_VoipSvcStatusInfo::VoipSvcStatus`

## 8.103 imsa\_VtRatInfo Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t VtRatVal`

### 8.103.1 Detailed Description

This structure hold parameters about VT RAT info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>VtRatVal</i>	<ul style="list-style-type: none"><li>• Values:<ul style="list-style-type: none"><li>• 0-IMSA_WLAN - IMS service is registered on WLAN</li><li>• 1-IMSA_WWAN - IMS service is registered on WWAN</li><li>• 2-IMSA_IWLAN - IMS service is registered on interworking WLAN</li></ul></li></ul>

### 8.103.2 Field Documentation

8.103.2.1 `uint8_t imsa_VtRatInfo::TlvPresent`

8.103.2.2 `uint32_t imsa_VtRatInfo::VtRatVal`

## 8.104 imsa\_VtSvcStatusInfo Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint32_t` [VtSvcStatus](#)

### 8.104.1 Detailed Description

This structure hold parameters about VT service info.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>VtSvcStatus</i>	<ul style="list-style-type: none"> <li>• Values: -0 - IMS VT service is not available -2 - IMS VT is in full service</li> </ul>

### 8.104.2 Field Documentation

8.104.2.1 `uint8_t imsa_VtSvcStatusInfo::TlvPresent`

8.104.2.2 `uint32_t imsa_VtSvcStatusInfo::VtSvcStatus`

## 8.105 ipv6AddressInfo Struct Reference

### Data Fields

- `uint8_t` [IPv6PrefixLen](#)
- `uint16_t` [IPAddressV6](#) [8]

### 8.105.1 Detailed Description

This structure contains the IPv6 address information

#### Parameters

<i>IPv6PrefixLen</i>	Length of the received IPv6 address <ul style="list-style-type: none"> <li>• 0xff - Invalid IPv6 address information.</li> </ul>
<i>IPAddressV6</i>	IPv6 address(in network byte order)

### 8.105.2 Field Documentation

8.105.2.1 uint16\_t ipv6AddressInfo::IPAddressV6[8]

8.105.2.2 uint8\_t ipv6AddressInfo::IPV6PrefixLen

## 8.106 LibPackGPRSRequestedQoS Struct Reference

### Data Fields

- uint32\_t [precedenceClass](#)
- uint32\_t [delayClass](#)
- uint32\_t [reliabilityClass](#)
- uint32\_t [peakThroughputClass](#)
- uint32\_t [meanThroughputClass](#)

### 8.106.1 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>

### 8.106.2 Field Documentation

8.106.2.1 uint32\_t LibPackGPRSRequestedQoS::delayClass

8.106.2.2 uint32\_t LibPackGPRSRequestedQoS::meanThroughputClass

8.106.2.3 uint32\_t LibPackGPRSRequestedQoS::peakThroughputClass

8.106.2.4 uint32\_t LibPackGPRSRequestedQoS::precedenceClass

8.106.2.5 uint32\_t LibPackGPRSRequestedQoS::reliabilityClass

## 8.107 LibPackPCOIDList Struct Reference

### Data Fields

- uint16\_t [PcoList](#) [10]

### 8.107.1 Detailed Description

This structure contains information about the PCOID List

#### Parameters

<i>PcoList</i>	PCOID for Max 10
----------------	------------------

### 8.107.2 Field Documentation

8.107.2.1 `uint16_t LibPackPCOIDList::PcoList[10]`

## 8.108 LibPackPDNThrottleTimer Struct Reference

### Data Fields

- `uint32_t ThrottleTimer` [10]

### 8.108.1 Detailed Description

This structure contains information about the PDN throttle timer

#### Parameters

<i>ThrottleTimer</i>	Throttle Timer for Max 10 PDN connections
----------------------	---

### 8.108.2 Field Documentation

8.108.2.1 `uint32_t LibPackPDNThrottleTimer::ThrottleTimer[10]`

## 8.109 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](#)
- uint8\_t \* [pSupportEmergencyCalls](#)

### 8.109.1 Detailed Description

This structure contains Input parameters of [unpack\\_wds\\_SLQSGetProfileSettings\\_t](#) 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 uint8_ts describing the profile</li> <li>• NULL pointer - Invalid data.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Requested QoS</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Minimum QoS</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Requested QoS</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pGPRS-MinimumQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Minimum QoS</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• User name</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>



<i>pPassword</i>	<ul style="list-style-type: none"> <li>• Password</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></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> <li>• Bit to check in ParamPresenceMask - <b>28</b></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> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> <li>• IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></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> <li>• Bit to check in ParamPresenceMask - <b>31</b></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> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> <li>Traffic Flow Template</li> <li>Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> <li>Traffic Flow Template</li> <li>Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>pPdpContext</i>	<ul style="list-style-type: none"> <li>PDP context number</li> <li>Bit to check in ParamPresenceMask - <b>37</b></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> <li>Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pPrimaryID</i>	<ul style="list-style-type: none"> <li>PDP context primary ID</li> <li>function SLQSGetProfileSettings() returns a default value 0xFF if this parameter is not returned by the device</li> <li>Bit to check in ParamPresenceMask - <b>39</b></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> <li>Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> <li>UMTS requested QoS with Signalling Indication flag</li> <li>Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>pUMTSMinQoS-SigInd</i>	<ul style="list-style-type: none"> <li>UMTS minimum QoS with Signalling Indication flag</li> <li>Bit to check in ParamPresenceMask - <b>42</b></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> <li>Bit to check in ParamPresenceMask - <b>43</b></li> </ul>

<i>pSecondaryDN-SIPv6addpref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>44</b></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> <li>• Bit to check in ParamPresenceMask - <b>45</b></li> </ul>
<i>pQosClassID</i>	<ul style="list-style-type: none"> <li>• 3GPP LTE QoS parameters</li> <li>• Bit to check in ParamPresenceMask - <b>46</b></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 SLQSGetProfile-Settings().</li> <li>• Bit to check in ParamPresenceMask - <b>47</b></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 SLQSGetProfile-Settings().</li> <li>• Bit to check in ParamPresenceMask - <b>48</b></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 SLQSGetProfile-Settings().</li> <li>• Bit to check in ParamPresenceMask - <b>49</b></li> </ul>
<i>pSupport-EmergencyCalls</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled</li> <li>• Can be queried to get current status</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - FALSE(default)</li> <li>– 1 - TRUE</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>54</b></li> </ul>

## 8.109.2 Field Documentation

- 8.109.2.1 `uint8_t*` `LibpackProfile3GPP::pAddrAllocPref`
- 8.109.2.2 `uint8_t*` `LibpackProfile3GPP::pAPNClass`
- 8.109.2.3 `uint8_t*` `LibpackProfile3GPP::pAPNDisabledFlag`
- 8.109.2.4 `uint8_t*` `LibpackProfile3GPP::pAPNName`
- 8.109.2.5 `uint16_t*` `LibpackProfile3GPP::pAPNnameSize`
- 8.109.2.6 `uint8_t*` `LibpackProfile3GPP::pAuthenticationPref`
- 8.109.2.7 `LibPackGPRSRequestedQoS*` `LibpackProfile3GPP::pGPRSMinimumQoS`
- 8.109.2.8 `LibPackGPRSRequestedQoS*` `LibpackProfile3GPP::pGPRSRequestedQos`
- 8.109.2.9 `uint8_t*` `LibpackProfile3GPP::pImCnFlag`
- 8.109.2.10 `uint32_t*` `LibpackProfile3GPP::pIPv4AddrPref`
- 8.109.2.11 `uint16_t*` `LibpackProfile3GPP::pIPv6AddPref`
- 8.109.2.12 `uint8_t*` `LibpackProfile3GPP::pPassword`
- 8.109.2.13 `uint16_t*` `LibpackProfile3GPP::pPasswordSize`
- 8.109.2.14 `uint8_t*` `LibpackProfile3GPP::pPcscfAddrUsingDhcp`
- 8.109.2.15 `uint8_t*` `LibpackProfile3GPP::pPcscfAddrUsingPCO`
- 8.109.2.16 `uint32_t*` `LibpackProfile3GPP::pPDNInactivTimeout`
- 8.109.2.17 `uint8_t*` `LibpackProfile3GPP::pPdpAccessConFlag`
- 8.109.2.18 `uint8_t*` `LibpackProfile3GPP::pPdpContext`
- 8.109.2.19 `uint8_t*` `LibpackProfile3GPP::pPdpDataCompType`
- 8.109.2.20 `uint8_t*` `LibpackProfile3GPP::pPdpHdrCompType`
- 8.109.2.21 `uint8_t*` `LibpackProfile3GPP::pPDPTtype`
- 8.109.2.22 `uint32_t*` `LibpackProfile3GPP::pPriDNSIPv4AddPref`
- 8.109.2.23 `uint16_t*` `LibpackProfile3GPP::pPriDNSIPv6addpref`
- 8.109.2.24 `uint8_t*` `LibpackProfile3GPP::pPrimaryID`
- 8.109.2.25 `uint8_t*` `LibpackProfile3GPP::pProfilename`
- 8.109.2.26 `uint16_t*` `LibpackProfile3GPP::pProfilenameSize`
- 8.109.2.27 `LibPackQosClassID*` `LibpackProfile3GPP::pQosClassID`

- 8.109.2.28 uint32\_t\* LibpackProfile3GPP::pSecDNSIPv4AddPref
- 8.109.2.29 uint16\_t\* LibpackProfile3GPP::pSecDNSIPv6addpref
- 8.109.2.30 uint8\_t\* LibpackProfile3GPP::pSecondaryFlag
- 8.109.2.31 uint8\_t\* LibpackProfile3GPP::pSupportEmergencyCalls
- 8.109.2.32 LibPackTFTIDParams\* LibpackProfile3GPP::pTFTID1Params
- 8.109.2.33 LibPackTFTIDParams\* LibpackProfile3GPP::pTFTID2Params
- 8.109.2.34 LibPackUMTSQoS\* LibpackProfile3GPP::pUMTSMinQoS
- 8.109.2.35 LibPackUMTSReqQoSSigInd\* LibpackProfile3GPP::pUMTSMinQoSsigInd
- 8.109.2.36 LibPackUMTSQoS\* LibpackProfile3GPP::pUMTSReqQoS
- 8.109.2.37 LibPackUMTSReqQoSSigInd\* LibpackProfile3GPP::pUMTSReqQoSsigInd
- 8.109.2.38 uint8\_t\* LibpackProfile3GPP::pUsername
- 8.109.2.39 uint16\_t\* LibpackProfile3GPP::pUsernameSize

## 8.110 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 \* [pIsPcscfAddressNedded](#)
- uint32\_t \* [pPrimaryV4DnsAddress](#)
- uint32\_t \* [pSecondaryV4DnsAddress](#)
- uint16\_t \* [pPriV6DnsAddress](#)
- uint16\_t \* [pSecV6DnsAddress](#)

- uint8\_t \* [pRATType](#)
- uint8\_t \* [pAPNEnabled3GPP2](#)
- uint32\_t \* [pPDNInactivTimeout3GPP2](#)
- uint8\_t \* [pAPNClass3GPP2](#)

### 8.110.1 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> <li>• Bit to check in ParamPresenceMask - <b>144</b></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> <li>• Bit to check in ParamPresenceMask - <b>145</b></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> <li>• Bit to check in ParamPresenceMask - <b>146</b></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> <li>• Bit to check in ParamPresenceMask - <b>147</b></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> <li>• Bit to check in ParamPresenceMask - <b>148</b></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> <li>• Bit to check in ParamPresenceMask - <b>149</b></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> <li>• Bit to check in ParamPresenceMask - <b>150</b></li> </ul>

<i>pLcpCreqRetry-Count</i>	<ul style="list-style-type: none"> <li>• LCP Configuration Request Retry Count</li> <li>• Bit to check in ParamPresenceMask - <b>151</b></li> </ul>
<i>plpcpCreqRetry-Count</i>	<ul style="list-style-type: none"> <li>• IPCP Configuration Request Retry Count</li> <li>• Bit to check in ParamPresenceMask - <b>152</b></li> </ul>
<i>pAuthRetry-Count</i>	<ul style="list-style-type: none"> <li>• Authentication Retry Count value</li> <li>• Bit to check in ParamPresenceMask - <b>153</b></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> <li>• Bit to check in ParamPresenceMask - <b>154</b></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> <li>• Bit to check in ParamPresenceMask - <b>155</b></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> <li>• Bit to check in ParamPresenceMask - <b>155</b></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> <li>• Bit to check in ParamPresenceMask - <b>156</b></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> <li>• Bit to check in ParamPresenceMask - <b>156</b></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> <li>• Bit to check in ParamPresenceMask - <b>157</b></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> <li>• Bit to check in ParamPresenceMask - <b>158</b></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> <li>• Bit to check in ParamPresenceMask - <b>159</b></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> <li>• Bit to check in ParamPresenceMask - <b>160</b></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> <li>• Bit to check in ParamPresenceMask - <b>161</b></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> <li>• Bit to check in ParamPresenceMask - <b>161</b></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> <li>• Bit to check in ParamPresenceMask - <b>162</b></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> <li>• Bit to check in ParamPresenceMask - <b>163</b></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> <li>Bit to check in ParamPresenceMask - <b>164</b></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> <li>Bit to check in ParamPresenceMask - <b>165</b></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> <li>Bit to check in ParamPresenceMask - <b>166</b></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> <li>Bit to check in ParamPresenceMask - <b>167</b></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 SLQSGetProfile-Settings().</li> <li>Bit to check in ParamPresenceMask - <b>168</b></li> </ul>
<i>pAPNEnabled3-GPP2</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 SLQSGetProfile-Settings().</li> <li>Bit to check in ParamPresenceMask - <b>169</b></li> </ul>
<i>pPDNInactiv-Timeout3GPP2</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 SLQSGetProfile-Settings().</li> <li>Bit to check in ParamPresenceMask - <b>170</b></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 SLQSGetProfileSettings().</li> <li>• Bit to check in ParamPresenceMask - <b>171</b></li> </ul>
-----------------------	---

## 8.110.2 Field Documentation

- 8.110.2.1 uint8\_t\* LibpackProfile3GPP2::pAllowLinger
- 8.110.2.2 uint8\_t\* LibpackProfile3GPP2::pAPNClass3GPP2
- 8.110.2.3 uint8\_t\* LibpackProfile3GPP2::pAPNEnabled3GPP2
- 8.110.2.4 uint8\_t\* LibpackProfile3GPP2::pApnString
- 8.110.2.5 uint16\_t\* LibpackProfile3GPP2::pApnStringSize
- 8.110.2.6 uint8\_t\* LibpackProfile3GPP2::pAppPriority
- 8.110.2.7 uint32\_t\* LibpackProfile3GPP2::pAppType
- 8.110.2.8 uint8\_t\* LibpackProfile3GPP2::pAuthPassword
- 8.110.2.9 uint16\_t\* LibpackProfile3GPP2::pAuthPasswordSize
- 8.110.2.10 uint8\_t\* LibpackProfile3GPP2::pAuthProtocol
- 8.110.2.11 uint8\_t\* LibpackProfile3GPP2::pAuthRetryCount
- 8.110.2.12 uint16\_t\* LibpackProfile3GPP2::pAuthTimeout
- 8.110.2.13 uint8\_t\* LibpackProfile3GPP2::pDataMode
- 8.110.2.14 uint8\_t\* LibpackProfile3GPP2::pDataRate
- 8.110.2.15 uint16\_t\* LibpackProfile3GPP2::plpcpAckTimeout
- 8.110.2.16 uint8\_t\* LibpackProfile3GPP2::plpcpCreqRetryCount
- 8.110.2.17 uint8\_t\* LibpackProfile3GPP2::plsPcscfAddressNedded
- 8.110.2.18 uint16\_t\* LibpackProfile3GPP2::pLcpAckTimeout
- 8.110.2.19 uint8\_t\* LibpackProfile3GPP2::pLcpCreqRetryCount
- 8.110.2.20 uint8\_t\* LibpackProfile3GPP2::pNegoDnsSrvrPref
- 8.110.2.21 uint32\_t\* LibpackProfile3GPP2::pPDNInactivTimeout3GPP2
- 8.110.2.22 uint8\_t\* LibpackProfile3GPP2::pPdnType

- 8.110.2.23 uint32\_t\* LibpackProfile3GPP2::pPppSessCloseTimer1x
- 8.110.2.24 uint32\_t\* LibpackProfile3GPP2::pPppSessCloseTimerDO
- 8.110.2.25 uint32\_t\* LibpackProfile3GPP2::pPrimaryV4DnsAddress
- 8.110.2.26 uint16\_t\* LibpackProfile3GPP2::pPriV6DnsAddress
- 8.110.2.27 uint8\_t\* LibpackProfile3GPP2::pRATType
- 8.110.2.28 uint32\_t\* LibpackProfile3GPP2::pSecondaryV4DnsAddress
- 8.110.2.29 uint16\_t\* LibpackProfile3GPP2::pSecV6DnsAddress
- 8.110.2.30 uint8\_t\* LibpackProfile3GPP2::pUserId
- 8.110.2.31 uint16\_t\* LibpackProfile3GPP2::pUserIdSize

## 8.111 LibpackProfile3GPPV2 Struct Reference

### Data Fields

- uint8\_t \* pProfilename
- uint16\_t \* pProfilenameSize
- uint8\_t \* pDPtype
- uint8\_t \* pDpHdrCompType
- uint8\_t \* pDpDataCompType
- 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 \* pDpAccessConFlag
- uint8\_t \* pPcscfAddrUsingDhcp
- uint8\_t \* pImCnFlag
- LibPackTFTIDParams \* pTFTID1Params
- LibPackTFTIDParams \* pTFTID2Params
- uint8\_t \* pDpContext
- 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
- uint64\_t \* pAPNBearer
- uint8\_t \* pSupportEmergencyCalls
- uint16\_t \* pOperatorPCOID
- uint16\_t \* pMcc
- LibPackProfileMnc \* pMnc
- uint16\_t \* pMaxPDN
- uint16\_t \* pMaxPDNTimer
- uint16\_t \* pPDNWaitTimer
- uint32\_t \* pAppUserData
- uint8\_t \* pRoamDisallowFlag
- uint8\_t \* pPDNDisconnectWaitTimer
- uint8\_t \* pDnsWithDHCPFlag
- uint32\_t \* pLteRoamPDPTYPE
- uint32\_t \* pUmtsRoamPDPTYPE
- uint8\_t \* pIWLANtoLTEHandoverFlag
- uint8\_t \* pLTEtoIWLANHandoverFlag
- LibPackPDNThrottleTimer \* pPDNThrottleTimer
- uint32\_t \* pOverridePDPTYPE
- LibPackPCOIDList \* pPCOIDList
- uint8\_t \* pMsisdnFlag
- uint8\_t \* pPersistFlag
- uint8\_t \* pClatFlag
- uint8\_t \* pIPv6DelegFlag

### 8.111.1 Detailed Description

This structure contains Input parameters of [unpack\\_wds\\_SLQSGetProfileSettings\\_t](#) 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 uint8_ts describing the profile</li> <li>• NULL pointer - Invalid data.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pAPNName</i>	<ul style="list-style-type: none"> <li>• Access point name</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pPriDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pSecDNSIPv4-AddPref</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pUMTSReqQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Requested QoS</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pUMTSMinQoS</i>	<ul style="list-style-type: none"> <li>• UMTS Minimum QoS</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pGPRS-RequestedQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Requested QoS</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pGPRS-MinimumQoS</i>	<ul style="list-style-type: none"> <li>• GPRS Minimum QoS</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>

<i>pUsername</i>	<ul style="list-style-type: none"> <li>• User name</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pPassword</i>	<ul style="list-style-type: none"> <li>• Password</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></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> <li>• Bit to check in ParamPresenceMask - <b>28</b></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> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pIPv4AddrPref</i>	<ul style="list-style-type: none"> <li>• IPv4 Address Preference</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></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> <li>• Bit to check in ParamPresenceMask - <b>31</b></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> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>pTFTID1Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pTFTID2Params</i>	<ul style="list-style-type: none"> <li>• Traffic Flow Template</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>pPdpContext</i>	<ul style="list-style-type: none"> <li>• PDP context number</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></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> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pPrimaryID</i>	<ul style="list-style-type: none"> <li>• PDP context primary ID</li> <li>• default value 0xFF if parameter not returned by the device</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></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> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>pUMTSReqQoS-SigInd</i>	<ul style="list-style-type: none"> <li>• UMTS requested QoS with Signalling Indication flag</li> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>pUMTSMinQoS-SigInd</i>	<ul style="list-style-type: none"> <li>• UMTS minimum QoS with Signalling Indication flag</li> <li>• Bit to check in ParamPresenceMask - <b>42</b></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> <li>Bit to check in ParamPresenceMask - <b>43</b></li> </ul>
<i>pSecondaryDNSIPv6addpref</i>	<ul style="list-style-type: none"> <li>Secondary DNS IPv6 address preference</li> <li>Bit to check in ParamPresenceMask - <b>44</b></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> <li>Bit to check in ParamPresenceMask - <b>45</b></li> </ul>
<i>pQosClassID</i>	<ul style="list-style-type: none"> <li>3GPP LTE QoS parameters</li> <li>Bit to check in ParamPresenceMask - <b>46</b></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>Bit to check in ParamPresenceMask - <b>47</b></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>Bit to check in ParamPresenceMask - <b>48</b></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>Bit to check in ParamPresenceMask - <b>49</b></li> </ul>
<i>pAPNBearer</i>	<ul style="list-style-type: none"> <li>Optional 8 Byte numeric APN bearer mask <ul style="list-style-type: none"> <li>0x0000000000000001 – GSM</li> <li>0x0000000000000002 – WCDMA</li> <li>0x0000000000000004 – LTE</li> <li>0x8000000000000000 – Any</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>53</b></li> </ul>



<i>pSupport-EmergencyCalls</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled</li> <li>• Can be queried to get current status</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - FALSE(default)</li> <li>– 1 - TRUE</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>54</b></li> </ul>
<i>pOperatorPCO-ID</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes value indicating container ID of this PCO</li> <li>• Bit to check in ParamPresenceMask - <b>55</b></li> </ul>
<i>pMcc</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes value indicating Mobile country code</li> <li>• Range 0-999</li> <li>• Bit to check in ParamPresenceMask - <b>56</b></li> </ul>
<i>pMnc</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes Mobile network code and 1 byte flag to indicate if MNC includes PCS digit</li> <li>• range 0-999</li> <li>• Bit to check in ParamPresenceMask - <b>57</b></li> </ul>
<i>pMaxPDN</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes value indicating Max PDN connections per time block</li> <li>• Default is 20</li> <li>• Range 0-1023</li> <li>• Bit to check in ParamPresenceMask - <b>58</b></li> </ul>
<i>pMaxPDNTimer</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes value indicating Max PDN connection timer</li> <li>• Default is 300 sec</li> <li>• Range 0-3600 sec</li> <li>• Bit to check in ParamPresenceMask - <b>59</b></li> </ul>
<i>pPDNWaitTimer</i>	<ul style="list-style-type: none"> <li>• Optional 2 bytes value indicating PDN request wait interval</li> <li>• Default is 0 sec</li> <li>• Range 0-1023 sec</li> <li>• Bit to check in ParamPresenceMask - <b>60</b></li> </ul>
<i>pAppUserData</i>	<ul style="list-style-type: none"> <li>• Optional 4 bytes value indicating user data ID in the profile</li> <li>• Bit to check in ParamPresenceMask - <b>61</b></li> </ul>
<i>pRoamDisallow-Flag</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte value indicating roaming disallowed flag is set or not</li> <li>• If flag is set as 1 UE is allowed to connect with APN while roaming</li> <li>• Bit to check in ParamPresenceMask - <b>62</b></li> </ul>

<i>pPDN-DisconnectWait-Timer</i>	<ul style="list-style-type: none"> <li>• Optional 1 bytes value indicating PDN disconnect wait interval</li> <li>• Range 0-255 minutes</li> <li>• Bit to check in ParamPresenceMask - <b>63</b></li> </ul>
<i>pDnsWithDHCP-Flag</i>	<ul style="list-style-type: none"> <li>• Optional 1 byte value indicating getting DNS address using DHCP <ul style="list-style-type: none"> <li>– 0 - Dont request DNS with DHCP</li> <li>– 1 - Request DNS address with DHCP</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>64</b></li> </ul>
<i>pLteRoamPDP-Type</i>	<ul style="list-style-type: none"> <li>• Optional 4 bytes value indicating LTE roaming PDP type <ul style="list-style-type: none"> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4 (0x00) – IPv4</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV6 (0x01) – IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4V6 (0x02) – IPv4 and IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_MAX (0xFF) – Nothing is configured</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>65</b></li> </ul>
<i>pUmtsRoamPDP-Type</i>	<ul style="list-style-type: none"> <li>• Optional 4 bytes value indicating UMTS roaming PDP type <ul style="list-style-type: none"> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4 (0x00) – IPv4</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV6 (0x01) – IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4V6 (0x02) – IPv4 and IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_MAX (0xFF) – Nothing is configured</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>66</b></li> </ul>
<i>pIWLANToLTE-HandoverFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate IWLAN to LTE handover is allowed or not <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Allowed</li> <li>– 0 -(FALSE) - Disallowed</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>67</b></li> </ul>
<i>pLTEtoIWLAN-HandoverFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate LTE to IWLAN handover is allowed or not <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Allowed</li> <li>– 0 -(FALSE) - Disallowed</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>68</b></li> </ul>
<i>pPDNThrottle-Timer</i>	<ul style="list-style-type: none"> <li>• Optional param for the throttle timer values for Max 10 PDN connection</li> <li>• Bit to check in ParamPresenceMask - <b>69</b></li> </ul>
<i>pOverridePDP-Type</i>	<ul style="list-style-type: none"> <li>• Optional 4 bytes value indicating overriding home PDP type <ul style="list-style-type: none"> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4 (0x00) – IPv4</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV6 (0x01) – IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_IPV4V6 (0x02) – IPv4 and IPv6</li> <li>– WDS_COMMON_PDP_TYPE_PDP_MAX (0xFF) – Nothing is configured</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>70</b></li> </ul>

<i>pPCOIDList</i>	<ul style="list-style-type: none"> <li>• Optional param with 20 bytes, List of 10 PCOs</li> <li>• Bit to check in ParamPresenceMask - <b>71</b></li> </ul>
<i>pMsisdnFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate MSISDN flag <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Enabled</li> <li>– 0 -(FALSE) - Disabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>72</b></li> </ul>
<i>pPersistFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate profile persistent flag <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Enabled</li> <li>– 0 -(FALSE) - Disabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>
<i>pClatFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate if CLAT is enabled or not <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Enabled</li> <li>– 0 -(FALSE) - Disabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>222</b></li> </ul>
<i>pIPv6DelegFlag</i>	<ul style="list-style-type: none"> <li>• This boolean value is used to indicate if IPV6 prefix delegation flag is enabled or not <ul style="list-style-type: none"> <li>– 1 -(TRUE) - Enabled</li> <li>– 0 -(FALSE) - Disabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>223</b></li> </ul>

## 8.111.2 Field Documentation

8.111.2.1 `uint8_t*` LibpackProfile3GPPV2::pAddrAllocPref

8.111.2.2 `uint64_t*` LibpackProfile3GPPV2::pAPNBearer

8.111.2.3 `uint8_t*` LibpackProfile3GPPV2::pAPNClass

8.111.2.4 `uint8_t*` LibpackProfile3GPPV2::pAPNDisabledFlag

8.111.2.5 `uint8_t*` LibpackProfile3GPPV2::pAPNName

8.111.2.6 `uint16_t*` LibpackProfile3GPPV2::pAPNnameSize

8.111.2.7 `uint32_t*` LibpackProfile3GPPV2::pAppUserData

8.111.2.8 `uint8_t*` LibpackProfile3GPPV2::pAuthenticationPref

8.111.2.9 `uint8_t*` LibpackProfile3GPPV2::pClatFlag

8.111.2.10 `uint8_t*` LibpackProfile3GPPV2::pDnsWithDHCPFlag

8.111.2.11 `LibPackGPRSRequestedQoS*` LibpackProfile3GPPV2::pGPRSMinimumQoS

- 8.111.2.12 **LibPackGPRSRequestedQoS\*** LibpackProfile3GPPV2::pGPRSRequestedQos
- 8.111.2.13 uint8\_t\* LibpackProfile3GPPV2::plmCnFlag
- 8.111.2.14 uint32\_t\* LibpackProfile3GPPV2::pIPv4AddrPref
- 8.111.2.15 uint16\_t\* LibpackProfile3GPPV2::pIPv6AddrPref
- 8.111.2.16 uint8\_t\* LibpackProfile3GPPV2::pIPv6DelegFlag
- 8.111.2.17 uint8\_t\* LibpackProfile3GPPV2::pIWLANtoLTEHandoverFlag
- 8.111.2.18 uint32\_t\* LibpackProfile3GPPV2::pLteRoamPDPTType
- 8.111.2.19 uint8\_t\* LibpackProfile3GPPV2::pLTEtoWLANHandoverFlag
- 8.111.2.20 uint16\_t\* LibpackProfile3GPPV2::pMaxPDN
- 8.111.2.21 uint16\_t\* LibpackProfile3GPPV2::pMaxPDNTimer
- 8.111.2.22 uint16\_t\* LibpackProfile3GPPV2::pMcc
- 8.111.2.23 **LibPackProfileMnc\*** LibpackProfile3GPPV2::pMnc
- 8.111.2.24 uint8\_t\* LibpackProfile3GPPV2::pMsisdnFlag
- 8.111.2.25 uint16\_t\* LibpackProfile3GPPV2::pOperatorPCOID
- 8.111.2.26 uint32\_t\* LibpackProfile3GPPV2::pOverridePDPTType
- 8.111.2.27 uint8\_t\* LibpackProfile3GPPV2::pPassword
- 8.111.2.28 uint16\_t\* LibpackProfile3GPPV2::pPasswordSize
- 8.111.2.29 **LibPackPCOIDList\*** LibpackProfile3GPPV2::pPCOIDList
- 8.111.2.30 uint8\_t\* LibpackProfile3GPPV2::pPcscfAddrUsingDhcp
- 8.111.2.31 uint8\_t\* LibpackProfile3GPPV2::pPcscfAddrUsingPCO
- 8.111.2.32 uint8\_t\* LibpackProfile3GPPV2::pPDNDisconnectWaitTimer
- 8.111.2.33 uint32\_t\* LibpackProfile3GPPV2::pPDNInactivTimeout
- 8.111.2.34 **LibPackPDNThrottleTimer\*** LibpackProfile3GPPV2::pPDNThrottleTimer
- 8.111.2.35 uint16\_t\* LibpackProfile3GPPV2::pPDNWaitTimer
- 8.111.2.36 uint8\_t\* LibpackProfile3GPPV2::pPdpAccessConFlag
- 8.111.2.37 uint8\_t\* LibpackProfile3GPPV2::pPdpContext
- 8.111.2.38 uint8\_t\* LibpackProfile3GPPV2::pPdpDataCompType
- 8.111.2.39 uint8\_t\* LibpackProfile3GPPV2::pPdpHdrCompType

- 8.111.2.40 `uint8_t*` LibpackProfile3GPPV2::pPDType
- 8.111.2.41 `uint8_t*` LibpackProfile3GPPV2::pPersistFlag
- 8.111.2.42 `uint32_t*` LibpackProfile3GPPV2::pPriDNSIPv4AddPref
- 8.111.2.43 `uint16_t*` LibpackProfile3GPPV2::pPriDNSIPv6addpref
- 8.111.2.44 `uint8_t*` LibpackProfile3GPPV2::pPrimaryID
- 8.111.2.45 `uint8_t*` LibpackProfile3GPPV2::pProfilename
- 8.111.2.46 `uint16_t*` LibpackProfile3GPPV2::pProfilenameSize
- 8.111.2.47 `LibPackQosClassID*` LibpackProfile3GPPV2::pQosClassID
- 8.111.2.48 `uint8_t*` LibpackProfile3GPPV2::pRoamDisallowFlag
- 8.111.2.49 `uint32_t*` LibpackProfile3GPPV2::pSecDNSIPv4AddPref
- 8.111.2.50 `uint16_t*` LibpackProfile3GPPV2::pSecDNSIPv6addpref
- 8.111.2.51 `uint8_t*` LibpackProfile3GPPV2::pSecondaryFlag
- 8.111.2.52 `uint8_t*` LibpackProfile3GPPV2::pSupportEmergencyCalls
- 8.111.2.53 `LibPackTFTIDParams*` LibpackProfile3GPPV2::pTFTID1Params
- 8.111.2.54 `LibPackTFTIDParams*` LibpackProfile3GPPV2::pTFTID2Params
- 8.111.2.55 `LibPackUMTSQoS*` LibpackProfile3GPPV2::pUMTSMinQoS
- 8.111.2.56 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPPV2::pUMTSMinQoSsigInd
- 8.111.2.57 `LibPackUMTSQoS*` LibpackProfile3GPPV2::pUMTSReqQoS
- 8.111.2.58 `LibPackUMTSReqQoSSigInd*` LibpackProfile3GPPV2::pUMTSReqQoSsigInd
- 8.111.2.59 `uint32_t*` LibpackProfile3GPPV2::pUmtsRoamPDType
- 8.111.2.60 `uint8_t*` LibpackProfile3GPPV2::pUsername
- 8.111.2.61 `uint16_t*` LibpackProfile3GPPV2::pUsernameSize

## 8.112 LibPackprofile\_3GPP Struct Reference

### Data Fields

- `uint8_t *` [pProfilename](#)
- `uint16_t *` [pProfilenameSize](#)
- `uint8_t *` [pPDType](#)
- `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
- uint8\_t \* pSupportEmergencyCalls

### 8.112.1 Detailed Description

This structure contains Input/Output parameters of pack\_wds\_SLQSCreateProfile

- 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 SLQSGetProfileSettings() 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 SLQSGetProfileSettings().</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 SLQSGetProfileSettings().</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 SLQSGetProfileSettings().</li> </ul>
<i>pSupport-EmergencyCalls</i>	<ul style="list-style-type: none"> <li>• Optional 1 Byte Flag indicating if the emergency call support is disabled/enabled</li> <li>• If set, the profile can be used for making emergency calls</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0 - FALSE(default)</li> <li>– 1 - TRUE</li> </ul> </li> </ul>

## 8.112.2 Field Documentation

8.112.2.1 uint8\_t\* LibPackprofile\_3GPP::pAddrAllocPref

8.112.2.2 uint8\_t\* LibPackprofile\_3GPP::pAPNClass

8.112.2.3 uint8\_t\* LibPackprofile\_3GPP::pAPNDisabledFlag

8.112.2.4 uint8\_t\* LibPackprofile\_3GPP::pAPNName

8.112.2.5 uint16\_t\* LibPackprofile\_3GPP::pAPNnameSize

8.112.2.6 uint8\_t\* LibPackprofile\_3GPP::pAuthenticationPref

8.112.2.7 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSMinimumQoS

8.112.2.8 LibPackGPRSRequestedQoS\* LibPackprofile\_3GPP::pGPRSRequestedQoS

8.112.2.9 uint8\_t\* LibPackprofile\_3GPP::plmCnFlag

8.112.2.10 uint32\_t\* LibPackprofile\_3GPP::pIPv4AddrPref

8.112.2.11 uint16\_t\* LibPackprofile\_3GPP::pIPv6AddPref

8.112.2.12 uint8\_t\* LibPackprofile\_3GPP::pPassword

8.112.2.13 uint16\_t\* LibPackprofile\_3GPP::pPasswordSize

8.112.2.14 uint8\_t\* LibPackprofile\_3GPP::pPcscfAddrUsingDhcp

8.112.2.15 uint8\_t\* LibPackprofile\_3GPP::pPcscfAddrUsingPCO

8.112.2.16 uint32\_t\* LibPackprofile\_3GPP::pPDNInactivTimeout

8.112.2.17 uint8\_t\* LibPackprofile\_3GPP::pPdpAccessConFlag

8.112.2.18 uint8\_t\* LibPackprofile\_3GPP::pPdpContext

8.112.2.19 uint8\_t\* LibPackprofile\_3GPP::pPdpDataCompType

- 8.112.2.20 `uint8_t*` `LibPackprofile_3GPP::pPdpHdrCompType`
- 8.112.2.21 `uint8_t*` `LibPackprofile_3GPP::pPDPTtype`
- 8.112.2.22 `uint32_t*` `LibPackprofile_3GPP::pPriDNSIPv4AddPref`
- 8.112.2.23 `uint16_t*` `LibPackprofile_3GPP::pPriDNSIPv6addpref`
- 8.112.2.24 `uint8_t*` `LibPackprofile_3GPP::pPrimaryID`
- 8.112.2.25 `uint8_t*` `LibPackprofile_3GPP::pProfilename`
- 8.112.2.26 `uint16_t*` `LibPackprofile_3GPP::pProfilenameSize`
- 8.112.2.27 `LibPackQosClassID*` `LibPackprofile_3GPP::pQosClassID`
- 8.112.2.28 `uint32_t*` `LibPackprofile_3GPP::pSecDNSIPv4AddPref`
- 8.112.2.29 `uint16_t*` `LibPackprofile_3GPP::pSecDNSIPv6addpref`
- 8.112.2.30 `uint8_t*` `LibPackprofile_3GPP::pSecondaryFlag`
- 8.112.2.31 `uint8_t*` `LibPackprofile_3GPP::pSupportEmergencyCalls`
- 8.112.2.32 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID1Params`
- 8.112.2.33 `LibPackTFTIDParams*` `LibPackprofile_3GPP::pTFTID2Params`
- 8.112.2.34 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSMinQoS`
- 8.112.2.35 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSMinQoSsigInd`
- 8.112.2.36 `LibPackUMTSQoS*` `LibPackprofile_3GPP::pUMTSReqQoS`
- 8.112.2.37 `LibPackUMTSReqQoSSigInd*` `LibPackprofile_3GPP::pUMTSReqQoSSigInd`
- 8.112.2.38 `uint8_t*` `LibPackprofile_3GPP::pUsername`
- 8.112.2.39 `uint16_t*` `LibPackprofile_3GPP::pUsernameSize`

## 8.113 LibPackprofile\_3GPP2 Struct Reference

### Data Fields

- `uint8_t*` `pNegoDnsSvrPref`
- `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 \* [pIsPcscfAddressNedded](#)
- uint32\_t \* [pPrimaryV4DnsAddress](#)
- uint32\_t \* [pSecondaryV4DnsAddress](#)
- uint16\_t \* [pPriV6DnsAddress](#)
- uint16\_t \* [pSecV6DnsAddress](#)
- uint8\_t \* [pRATType](#)
- uint8\_t \* [pAPNEnabled3GPP2](#)
- uint32\_t \* [pPDNInactivTimeout3GPP2](#)
- uint8\_t \* [pAPNClass3GPP2](#)

### 8.113.1 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 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 SLQSGetProfile-Settings().</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 SLQSGetProfile-Settings().</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 SLQSGetProfile-Settings().</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 SLQSGetProfile-Settings().</li> </ul>

## 8.113.2 Field Documentation

8.113.2.1 uint8\_t\* LibPackprofile\_3GPP2::pAllowLinger

8.113.2.2 uint8\_t\* LibPackprofile\_3GPP2::pAPNClass3GPP2

8.113.2.3 uint8\_t\* LibPackprofile\_3GPP2::pAPNEnabled3GPP2

8.113.2.4 uint8\_t\* LibPackprofile\_3GPP2::pApnString

- 8.113.2.5    `uint16_t*` `LibPackprofile_3GPP2::pApnStringSize`
- 8.113.2.6    `uint8_t*` `LibPackprofile_3GPP2::pAppPriority`
- 8.113.2.7    `uint32_t*` `LibPackprofile_3GPP2::pAppType`
- 8.113.2.8    `uint8_t*` `LibPackprofile_3GPP2::pAuthPassword`
- 8.113.2.9    `uint16_t*` `LibPackprofile_3GPP2::pAuthPassword_tSize`
- 8.113.2.10   `uint8_t*` `LibPackprofile_3GPP2::pAuthProtocol`
- 8.113.2.11   `uint8_t*` `LibPackprofile_3GPP2::pAuthRetryCount`
- 8.113.2.12   `uint16_t*` `LibPackprofile_3GPP2::pAuthTimeout`
- 8.113.2.13   `uint8_t*` `LibPackprofile_3GPP2::pDataMode`
- 8.113.2.14   `uint8_t*` `LibPackprofile_3GPP2::pDataRate`
- 8.113.2.15   `uint16_t*` `LibPackprofile_3GPP2::plpcpAckTimeout`
- 8.113.2.16   `uint8_t*` `LibPackprofile_3GPP2::plpcpCreqRetryCount`
- 8.113.2.17   `uint8_t*` `LibPackprofile_3GPP2::plsPcscfAddressNedded`
- 8.113.2.18   `uint16_t*` `LibPackprofile_3GPP2::pLcpAckTimeout`
- 8.113.2.19   `uint8_t*` `LibPackprofile_3GPP2::pLcpCreqRetryCount`
- 8.113.2.20   `uint8_t*` `LibPackprofile_3GPP2::pNegoDnsSrvrPref`
- 8.113.2.21   `uint32_t*` `LibPackprofile_3GPP2::pPDNInactivTimeout3GPP2`
- 8.113.2.22   `uint8_t*` `LibPackprofile_3GPP2::pPdnType`
- 8.113.2.23   `uint32_t*` `LibPackprofile_3GPP2::pPppSessCloseTimer1x`
- 8.113.2.24   `uint32_t*` `LibPackprofile_3GPP2::pPppSessCloseTimerDO`
- 8.113.2.25   `uint32_t*` `LibPackprofile_3GPP2::pPrimaryV4DnsAddress`
- 8.113.2.26   `uint16_t*` `LibPackprofile_3GPP2::pPriV6DnsAddress`
- 8.113.2.27   `uint8_t*` `LibPackprofile_3GPP2::pRATType`
- 8.113.2.28   `uint32_t*` `LibPackprofile_3GPP2::pSecondaryV4DnsAddress`
- 8.113.2.29   `uint16_t*` `LibPackprofile_3GPP2::pSecV6DnsAddress`
- 8.113.2.30   `uint8_t*` `LibPackprofile_3GPP2::pUserId`
- 8.113.2.31   `uint16_t*` `LibPackprofile_3GPP2::pUserIdSize`

## 8.114    **LibPackProfileMnc Struct Reference**



## Data Fields

- uint16\_t [MNC](#)
- uint8\_t [PCSFlag](#)

### 8.114.1 Detailed Description

This structure contains the MNC Information

#### Parameters

<i>MNC</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code</li> <li>• range 0-999</li> </ul>
<i>PCSFlag</i>	<ul style="list-style-type: none"> <li>• Indicate if PCS flag is included <ul style="list-style-type: none"> <li>– 0 - FALSE</li> <li>– 1 - TRUE</li> </ul> </li> </ul>

### 8.114.2 Field Documentation

8.114.2.1 uint16\_t LibPackProfileMnc::MNC

8.114.2.2 uint8\_t LibPackProfileMnc::PCSFlag

## 8.115 LibPackQosClassID Struct Reference

## Data Fields

- uint8\_t [QCI](#)
- uint32\_t [gDIBitRate](#)
- uint32\_t [maxDIBitRate](#)
- uint32\_t [gUIBitRate](#)
- uint32\_t [maxUIBitRate](#)

### 8.115.1 Detailed Description

This 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>maxDlBitRate</i>	<ul style="list-style-type: none"> <li>Maximum DL bit rate</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>

### 8.115.2 Field Documentation

8.115.2.1 `uint32_t LibPackQosClassID::gDlBitRate`

8.115.2.2 `uint32_t LibPackQosClassID::gUlBitRate`

8.115.2.3 `uint32_t LibPackQosClassID::maxDlBitRate`

8.115.2.4 `uint32_t LibPackQosClassID::maxUlBitRate`

8.115.2.5 `uint8_t LibPackQosClassID::QCI`

## 8.116 LibPackTFTIDParams Struct Reference

### Data Fields

- `uint8_t filterId`
- `uint8_t eValid`
- `uint8_t ipVersion`
- `uint16_t * pSourceIP`
- `uint8_t sourceIPMask`
- `uint8_t nextHeader`
- `uint16_t destPortRangeStart`
- `uint16_t destPortRangeEnd`
- `uint16_t srcPortRangeStart`
- `uint16_t srcPortRangeEnd`
- `uint32_t IPSECSPi`
- `uint16_t tosMask`
- `uint32_t flowLabel`

### 8.116.1 Detailed Description

This 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>ipVersion</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>pSourceIP</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>

## 8.116.2 Field Documentation

8.116.2.1 `uint16_t LibPackTFTIDParams::destPortRangeEnd`

8.116.2.2 `uint16_t LibPackTFTIDParams::destPortRangeStart`

8.116.2.3 `uint8_t LibPackTFTIDParams::eValid`

8.116.2.4 `uint8_t LibPackTFTIDParams::filterId`

8.116.2.5 `uint32_t LibPackTFTIDParams::flowLabel`

- 8.116.2.6    `uint32_t LibPackTFTIDParams::IPSECSPI`
- 8.116.2.7    `uint8_t LibPackTFTIDParams::ipVersion`
- 8.116.2.8    `uint8_t LibPackTFTIDParams::nextHeader`
- 8.116.2.9    `uint16_t* LibPackTFTIDParams::pSourceIP`
- 8.116.2.10   `uint8_t LibPackTFTIDParams::sourceIPMask`
- 8.116.2.11   `uint16_t LibPackTFTIDParams::srcPortRangeEnd`
- 8.116.2.12   `uint16_t LibPackTFTIDParams::srcPortRangeStart`
- 8.116.2.13   `uint16_t LibPackTFTIDParams::tosMask`

## 8.117    **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](#)

### 8.117.1    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> <li>• 0xff - Invalid UMTS Quality Of Service Information.</li> </ul>
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Maximum uplink bit rate in bits/sec</li> </ul>

<i>maxDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Maximum downlink bit rate in bits/sec</li> </ul>
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed uplink bit rate in bits/sec</li> </ul>
<i>grntDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Guranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDelivery-Order</i>	<ul style="list-style-type: none"> <li>- Qos delivery order</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - delivery order on</li> <li>• 0x02 - delivery order off</li> </ul>
<i>maxSDUSize</i>	<ul style="list-style-type: none"> <li>• Maximum SDU size</li> </ul>
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> <li>- SDU error ratio</li> <li>• Target value for fraction of SDUs lost or detected as erroneous.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>1 \cdot 10^{-2}</math></li> <li>• 0x02 - <math>7 \cdot 10^{-3}</math></li> <li>• 0x03 - <math>1 \cdot 10^{-3}</math></li> <li>• 0x04 - <math>1 \cdot 10^{-4}</math></li> <li>• 0x05 - <math>1 \cdot 10^{-5}</math></li> <li>• 0x06 - <math>1 \cdot 10^{-6}</math></li> <li>• 0x07 - <math>1 \cdot 10^{-1}</math></li> </ul>
<i>resBerRatio</i>	<ul style="list-style-type: none"> <li>- Residual bit error ratio</li> <li>• Target value for undetected bit error ratio in in the delivered SDUs.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>5 \cdot 10^{-2}</math></li> <li>• 0x02 - <math>1 \cdot 10^{-2}</math></li> <li>• 0x03 - <math>5 \cdot 10^{-3}</math></li> <li>• 0x04 - <math>4 \cdot 10^{-3}</math></li> <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>	- Delivery of erroneous SDUs <ul style="list-style-type: none"> <li>• Indicates whether SDUs detected as erroneous shall be delivered or not.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>5 \cdot 10^{-2}</math></li> <li>• 0x02 - <math>1 \cdot 10^{-2}</math></li> <li>• 0x03 - <math>5 \cdot 10^{-3}</math></li> <li>• 0x04 - <math>4 \cdot 10^{-3}</math></li> <li>• 0x05 - <math>1 \cdot 10^{-3}</math></li> <li>• 0x06 - <math>1 \cdot 10^{-4}</math></li> <li>• 0x07 - <math>1 \cdot 10^{-5}</math></li> <li>• 0x08 - <math>1 \cdot 10^{-6}</math></li> <li>• 0x09 - <math>1 \cdot 10^{-8}</math></li> </ul>
<i>transferDelay</i>	- Transfer delay (ms) <ul style="list-style-type: none"> <li>• Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.</li> </ul>
<i>trafficPriority</i>	- Transfer handling priority <ul style="list-style-type: none"> <li>• Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

#### Note

Check [trafficClass](#) before use.

### 8.117.2 Field Documentation

- 8.117.2.1 `uint8_t LibPackUMTSQoS::deliveryErrSDU`
- 8.117.2.2 `uint32_t LibPackUMTSQoS::grntDownlinkBitrate`
- 8.117.2.3 `uint32_t LibPackUMTSQoS::grntUplinkBitrate`
- 8.117.2.4 `uint32_t LibPackUMTSQoS::maxDownlinkBitrate`
- 8.117.2.5 `uint32_t LibPackUMTSQoS::maxSDUSize`
- 8.117.2.6 `uint32_t LibPackUMTSQoS::maxUplinkBitrate`
- 8.117.2.7 `uint8_t LibPackUMTSQoS::qosDeliveryOrder`
- 8.117.2.8 `uint8_t LibPackUMTSQoS::resBerRatio`
- 8.117.2.9 `uint8_t LibPackUMTSQoS::sduErrorRatio`
- 8.117.2.10 `uint8_t LibPackUMTSQoS::trafficClass`
- 8.117.2.11 `uint32_t LibPackUMTSQoS::trafficPriority`
- 8.117.2.12 `uint32_t LibPackUMTSQoS::transferDelay`

## 8.118 LibPackUMTSReqQoS SigInd Struct Reference

### Data Fields

- [LibPackUMTSQoS UMTSReqQoS](#)
- [uint8\\_t SigInd](#)

### 8.118.1 Detailed Description

This 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> <li>• See <a href="#">LibPackUMTSQoS</a></li> </ul>
<i>SigInd</i>	- Signaling Indication flag <ul style="list-style-type: none"> <li>• TRUE - Signaling indication ON</li> <li>• FALSE - Signaling indication OFF</li> </ul>

### 8.118.2 Field Documentation

8.118.2.1 [uint8\\_t LibPackUMTSReqQoS SigInd::SigInd](#)

8.118.2.2 [LibPackUMTSQoS LibPackUMTSReqQoS SigInd::UMTSReqQoS](#)

## 8.119 loc\_accelAcceptReady Struct Reference

### Data Fields

- [uint8\\_t injectEnable](#)
- [uint16\\_t samplesPerBatch](#)
- [uint16\\_t batchPerSec](#)

### 8.119.1 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               <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>• <i>samplesPerBatch</i> must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>• LTE NAS version minor</li> <li>• Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>• <i>BatchesPerSecond</i> must be a nonzero positive value.</li> </ul>

## 8.119.2 Field Documentation

8.119.2.1 `uint16_t loc_accelAcceptReady::batchPerSec`

8.119.2.2 `uint8_t loc_accelAcceptReady::injectEnable`

8.119.2.3 `uint16_t loc_accelAcceptReady::samplesPerBatch`

## 8.120 loc\_accelTempAcceptReady Struct Reference

### Data Fields

- `uint8_t injectEnable`
- `uint16_t samplesPerBatch`
- `uint16_t batchPerSec`

### 8.120.1 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 <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>• <i>samplesPerBatch</i> must be a nonzero positive value.</li> </ul>
<i>batchPerSec</i>	<ul style="list-style-type: none"> <li>• LTE NAS version minor</li> <li>• Number of sensor-data batches the GNSS location engine is to receive per second.</li> <li>• <i>BatchesPerSecond</i> must be a nonzero positive value.</li> </ul>



## 8.120.2 Field Documentation

8.120.2.1 uint16\_t loc\_accelTempAcceptReady::batchPerSec

8.120.2.2 uint8\_t loc\_accelTempAcceptReady::injectEnable

8.120.2.3 uint16\_t loc\_accelTempAcceptReady::samplesPerBatch

## 8.121 loc\_AppProviderInfoTlv Struct Reference

### Data Fields

- uint8\_t [providerLen](#)
- char [provider](#) [255]
- uint8\_t [nameLen](#)
- char [name](#) [255]
- uint8\_t [verValid](#)
- uint8\_t [versionLen](#)
- char [version](#) [255]
- uint8\_t [TlvPresent](#)

### 8.121.1 Detailed Description

Contain fields in struct [loc\\_AppProviderInfoTlv](#)

#### Parameters

<i>providerLen</i>	<ul style="list-style-type: none"> <li>• number of sets of the following elements: <ul style="list-style-type: none"> <li>– provider</li> </ul> </li> </ul>
<i>provider</i>	<ul style="list-style-type: none"> <li>• Application provider.</li> </ul>
<i>nameLen</i>	<ul style="list-style-type: none"> <li>• number of sets of the following elements: <ul style="list-style-type: none"> <li>– name</li> </ul> </li> </ul>
<i>name</i>	<ul style="list-style-type: none"> <li>• Application name.</li> </ul>
<i>verValid</i>	<ul style="list-style-type: none"> <li>• Specifies whether the application version string contains a valid value: 0x00 (FALSE) – Application version string is invalid 0x01 (TRUE) – Application version string is valid</li> </ul>
<i>versionLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements: version</li> </ul>
<i>version</i>	<ul style="list-style-type: none"> <li>• Application version.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

## Note

None

## 8.121.2 Field Documentation

8.121.2.1 char loc\_AppProviderInfoTlv::name[255]

8.121.2.2 uint8\_t loc\_AppProviderInfoTlv::nameLen

8.121.2.3 char loc\_AppProviderInfoTlv::provider[255]

8.121.2.4 uint8\_t loc\_AppProviderInfoTlv::providerLen

8.121.2.5 uint8\_t loc\_AppProviderInfoTlv::TlvPresent

8.121.2.6 char loc\_AppProviderInfoTlv::version[255]

8.121.2.7 uint8\_t loc\_AppProviderInfoTlv::versionLen

8.121.2.8 uint8\_t loc\_AppProviderInfoTlv::verValid

## 8.122 loc\_BdsSV Struct Reference

## Data Fields

- uint16\_t [id](#)
- uint8\_t [mask](#)

## 8.122.1 Detailed Description

This structure contains the BDS SV Info

## Parameters

<i>id</i>	<ul style="list-style-type: none"> <li>• SV 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>

## 8.122.2 Field Documentation

8.122.2.1 uint16\_t loc\_BdsSV::id

8.122.2.2 uint8\_t loc\_BdsSV::mask

## 8.123 loc\_BdsSVInfo Struct Reference

### Data Fields

- [uint8\\_t len](#)
- [loc\\_BdsSV \\* pSV](#)

### 8.123.1 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>

### 8.123.2 Field Documentation

8.123.2.1 [uint8\\_t loc\\_BdsSVInfo::len](#)

8.123.2.2 [loc\\_BdsSV\\* loc\\_BdsSVInfo::pSV](#)

## 8.124 loc\_CellDb Struct Reference

### Data Fields

- [uint32\\_t mask](#)

### 8.124.1 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>
-------------	--

## 8.124.2 Field Documentation

8.124.2.1 uint32\_t loc\_CellDb::mask

## 8.125 loc\_ClkInfo Struct Reference

## Data Fields

- uint32\_t [mask](#)

## 8.125.1 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>
-------------	--

## 8.125.2 Field Documentation

### 8.125.2.1 uint32\_t loc\_ClkInfo::mask

## 8.126 loc\_FixCriteriaStatusTlv Struct Reference

### Data Fields

- uint32\_t [status](#)
- uint8\_t [TlvPresent](#)

### 8.126.1 Detailed Description

Contain fields in struct [loc\\_FixCriteriaStatusTlv](#)

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Status of the get fix criteria 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>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

### 8.126.2 Field Documentation

8.126.2.1 `uint32_t loc_FixCriteriaStatusTlv::status`

8.126.2.2 `uint8_t loc_FixCriteriaStatusTlv::TlvPresent`

## 8.127 loc\_GnssData Struct Reference

### Data Fields

- `uint64_t mask`

### 8.127.1 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 SV 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 SV blacklist GPS</li> <li>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_GLO (0x01000000) - Mask to delete GNSS SV blacklist GLO</li> <li>– QMI_LOC_MASK_DELETE_GNSS_SV_BLACKLIST_BDS (0x02000000) - Mask to delete GNSS SV blacklist BDS</li> </ul> </li> </ul>
-------------	---

## 8.127.2 Field Documentation

## 8.127.2.1 uint64\_t loc\_GnssData::mask

## 8.128 loc\_gpsTime Struct Reference

## Data Fields

- uint16\_t [gpsWeek](#)
- uint32\_t [gpsTimeOfWeekMs](#)

### 8.128.1 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>

### 8.128.2 Field Documentation

8.128.2.1 uint32\_t loc\_gpsTime::gpsTimeOfWeekMs

8.128.2.2 uint16\_t loc\_gpsTime::gpsWeek

## 8.129 loc\_gyroAcceptReady Struct Reference

## Data Fields

- uint8\_t [injectEnable](#)
- uint16\_t [samplesPerBatch](#)
- uint16\_t [batchPerSec](#)

### 8.129.1 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>samplesPer-Batch</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>
--------------------	---

## 8.129.2 Field Documentation

8.129.2.1 uint16\_t loc\_gyroAcceptReady::batchPerSec

8.129.2.2 uint8\_t loc\_gyroAcceptReady::injectEnable

8.129.2.3 uint16\_t loc\_gyroAcceptReady::samplesPerBatch

## 8.130 loc\_gyroTempAcceptReady Struct Reference

### Data Fields

- uint8\_t [injectEnable](#)
- uint16\_t [samplesPerBatch](#)
- uint16\_t [batchPerSec](#)

### 8.130.1 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>

## 8.130.2 Field Documentation

8.130.2.1 uint16\_t loc\_gyroTempAcceptReady::batchPerSec

8.130.2.2 uint8\_t loc\_gyroTempAcceptReady::injectEnable

8.130.2.3 `uint16_t loc_gyroTempAcceptReady::samplesPerBatch`

## 8.131 `loc_HorAccuracyLvITlv` Struct Reference

### Data Fields

- `uint32_t` [accuracy](#)
- `uint8_t` [TlvPresent](#)

#### 8.131.1 Detailed Description

Contain fields in struct [loc\\_HorAccuracyLvITlv](#)

#### Parameters

<i>accuracy</i>	<ul style="list-style-type: none"> <li>• Horizontal accuracy level</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 0x00000001 – LOW: Client requires low horizontal accuracy</li> <li>– 0x00000002 – MED: Client requires medium horizontal accuracy</li> <li>– 0x00000003 – HIGH: Client requires high horizontal accuracy</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

#### 8.131.2 Field Documentation

8.131.2.1 `uint32_t loc_HorAccuracyLvITlv::accuracy`

8.131.2.2 `uint8_t loc_HorAccuracyLvITlv::TlvPresent`

## 8.132 `loc_IntermediateRptStateTlv` Struct Reference

### Data Fields

- `uint32_t` [intermediate](#)
- `uint8_t` [TlvPresent](#)

#### 8.132.1 Detailed Description

Contain fields in struct [loc\\_IntermediateRptStateTlv](#)

#### Parameters

<i>intermediate</i>	<ul style="list-style-type: none"> <li>• Intermediate Report state (ON, OFF). The client must explicitly set this field to OFF to stop receiving intermediate position reports. Intermediate position reports are generated at 1 Hz and are ON by default. If intermediate reports are turned ON, the client receives position reports even if the accuracy criteria is not met. The status in the position report is set to IN_PROGRESS for intermediate reports.</li> <li>• Valid values: 0x00000001 – ON: Client is interested in receiving intermediate reports 0x00000002 – OFF: Client is not interested in receiving intermediate reports</li> </ul>
---------------------	---

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>
-------------------	--

### 8.132.2 Field Documentation

8.132.2.1 uint32\_t loc\_IntermediateRptStateTlv::intermediate

8.132.2.2 uint8\_t loc\_IntermediateRptStateTlv::TlvPresent

## 8.133 loc\_IPv4Config Struct Reference

### Data Fields

- uint32\_t [IPv4Addr](#)
- uint16\_t [IPv4Port](#)

### 8.133.1 Detailed Description

This structure specifies information regarding the IPv4 address and port.

#### Parameters

<i>IPv4Addr</i>	<ul style="list-style-type: none"> <li>• IPv4 address.</li> </ul>
<i>IPv4Port</i>	<ul style="list-style-type: none"> <li>• IPv4 port.</li> </ul>

### 8.133.2 Field Documentation

8.133.2.1 uint32\_t loc\_IPv4Config::IPv4Addr

8.133.2.2 uint16\_t loc\_IPv4Config::IPv4Port

## 8.134 loc\_IPv4Info Struct Reference

### Data Fields

- uint32\_t [address](#)
- uint16\_t [port](#)
- uint8\_t [TlvPresent](#)

### 8.134.1 Detailed Description

This structure contains LOC Server IPV4 info field.

## Parameters

<i>address</i>	<ul style="list-style-type: none"> <li>IPv4 address.</li> </ul>
<i>port</i>	<ul style="list-style-type: none"> <li>IPv4 port.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present or not</li> </ul>

## 8.134.2 Field Documentation

8.134.2.1 uint32\_t loc\_IPv4Info::address

8.134.2.2 uint16\_t loc\_IPv4Info::port

8.134.2.3 uint8\_t loc\_IPv4Info::TlvPresent

## 8.135 loc\_IPv6Config Struct Reference

## Data Fields

- uint16\_t [IPv6Addr](#) [8]
- uint32\_t [IPv6Port](#)

## 8.135.1 Detailed Description

This structure specifies information regarding the IPv6 address and port.

## Parameters

<i>IPv6Addr</i>	<ul style="list-style-type: none"> <li>IPv6 address.</li> <li>Type - Array of unsigned integers</li> <li>Maximum length of the array: 8</li> </ul>
<i>IPv6Port</i>	<ul style="list-style-type: none"> <li>IPv6 port.</li> </ul>

## 8.135.2 Field Documentation

8.135.2.1 uint16\_t loc\_IPv6Config::IPv6Addr[8]

8.135.2.2 uint32\_t loc\_IPv6Config::IPv6Port

## 8.136 loc\_IPv6Info Struct Reference

## Data Fields

- uint16\_t [address](#) [8]

- uint32\_t [port](#)
- uint8\_t [TlvPresent](#)

### 8.136.1 Detailed Description

This structure contains LOC Get Server IPV6 info field.

#### Parameters

<i>address</i>	<ul style="list-style-type: none"> <li>• IPv6 address.</li> <li>• Type - Array of unsigned integers</li> <li>• Maximum length of the array - 8</li> </ul>
<i>port</i>	<ul style="list-style-type: none"> <li>• IPv6 port.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

### 8.136.2 Field Documentation

8.136.2.1 uint16\_t loc\_IPv6Info::address[8]

8.136.2.2 uint32\_t loc\_IPv6Info::port

8.136.2.3 uint8\_t loc\_IPv6Info::TlvPresent

## 8.137 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](#)

### 8.137.1 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>

### 8.137.2 Field Documentation

8.137.2.1 `uint8_t loc_LocApplicationInfo::appNameLength`

8.137.2.2 `uint8_t loc_LocApplicationInfo::appProviderLength`

8.137.2.3 `uint8_t loc_LocApplicationInfo::appVersionLength`

8.137.2.4 `uint8_t loc_LocApplicationInfo::appVersionValid`

8.137.2.5 `uint8_t* loc_LocApplicationInfo::pAppName`

8.137.2.6 `uint8_t* loc_LocApplicationInfo::pAppProvider`

8.137.2.7 `uint8_t* loc_LocApplicationInfo::pAppVersion`

## 8.138 loc\_MinIntervalTlv Struct Reference

### Data Fields

- `uint32_t interval`
- `uint8_t TlvPresent`

### 8.138.1 Detailed Description

Contain fields in struct `loc_MinIntervalTlv`

#### Parameters

<i>interval</i>	<ul style="list-style-type: none"> <li>Minimum Interval Between fixes</li> <li>Units: Milliseconds</li> </ul>
-----------------	---

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>
-------------------	--

### 8.138.2 Field Documentation

8.138.2.1 uint32\_t loc\_MinIntervalTlv::interval

8.138.2.2 uint8\_t loc\_MinIntervalTlv::TlvPresent

## 8.139 loc\_precisionDilution Struct Reference

### Data Fields

- uint32\_t [PDOP](#)
- uint32\_t [HDOP](#)
- uint32\_t [VDOP](#)

### 8.139.1 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>

### 8.139.2 Field Documentation

8.139.2.1 uint32\_t loc\_precisionDilution::HDOP

8.139.2.2 uint32\_t loc\_precisionDilution::PDOP

8.139.2.3 uint32\_t loc\_precisionDilution::VDOP

## 8.140 loc\_satelliteInfo Struct Reference

### Data Fields

- uint8\_t [svListLen](#)

- uint32\_t [validMask](#)
- uint32\_t [system](#)
- uint16\_t [gnssSvid](#)
- uint8\_t [healthStatus](#)
- uint32\_t [svStatus](#)
- uint8\_t [svInfoMask](#)
- float [elevation](#)
- float [azimuth](#)
- float [snr](#)

### 8.140.1 Detailed Description

Contain fields in struct [loc\\_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 SV 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 SV 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>SV process status. Valid values: <ul style="list-style-type: none"> <li>eQMI_LOC_SV_STATUS_IDLE (1) - SV is not being actively processed</li> <li>eQMI_LOC_SV_STATUS_SEARCH (2) - The system is searching for this SV</li> <li>eQMI_LOC_SV_STATUS_TRACK (3) - SV 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>

## 8.140.2 Field Documentation

8.140.2.1 float loc\_satelliteInfo::azimuth

8.140.2.2 float loc\_satelliteInfo::elevation

8.140.2.3 uint16\_t loc\_satelliteInfo::gnssSvId

8.140.2.4 uint8\_t loc\_satelliteInfo::healthStatus

8.140.2.5 float loc\_satelliteInfo::snr

8.140.2.6 `uint8_t loc_satelliteInfo::svInfoMask`

8.140.2.7 `uint8_t loc_satelliteInfo::svListLen`

8.140.2.8 `uint32_t loc_satelliteInfo::svStatus`

8.140.2.9 `uint32_t loc_satelliteInfo::system`

8.140.2.10 `uint32_t loc_satelliteInfo::validMask`

## 8.141 `loc_sensorDataUsage` Struct Reference

### Data Fields

- `uint32_t usageMask`
- `uint32_t aidingIndicatorMask`

### 8.141.1 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.             <ul style="list-style-type: none"> <li>– Value                 <ul style="list-style-type: none"> <li>* 0x00000001 - Accelerometer used</li> <li>* 0x00000002 - Gyroscope used</li> </ul> </li> </ul> </li> </ul>
<i>aidingIndicatorMask</i>	<ul style="list-style-type: none"> <li>• Specifies which results were aided by sensors.             <ul style="list-style-type: none"> <li>– Value                 <ul style="list-style-type: none"> <li>* 0x00000001 - AIDED_HEADING</li> <li>* 0x00000002 - AIDED_SPEED</li> <li>* 0x00000004 - AIDED_POSITION</li> <li>* 0x00000008 - AIDED_VELOCITY</li> </ul> </li> </ul> </li> </ul>

### 8.141.2 Field Documentation

8.141.2.1 `uint32_t loc_sensorDataUsage::aidingIndicatorMask`

8.141.2.2 `uint32_t loc_sensorDataUsage::usageMask`

## 8.142 `loc_SV` Struct Reference

### Data Fields

- `uint16_t id`
- `uint32_t system`
- `uint8_t mask`

### 8.142.1 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 <a href="#">loc_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>

### 8.142.2 Field Documentation

8.142.2.1 `uint16_t loc_SV::id`

8.142.2.2 `uint8_t loc_SV::mask`

8.142.2.3 `uint32_t loc_SV::system`

## 8.143 loc\_SVInfo Struct Reference

### Data Fields

- `uint8_t len`
- `loc_SV * pSV`

### 8.143.1 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>

## 8.143.2 Field Documentation

8.143.2.1 `uint8_t loc_SVInfo::len`8.143.2.2 `loc_SV* loc_SVInfo::pSV`8.144 `loc_svUsedforFix` Struct Reference

## Data Fields

- `uint8_t gnssSvUsedList_len`
- `uint16_t gnssSvUsedList [255]`

## 8.144.1 Detailed Description

This structure contains SVs Used to Calculate the Fix.

## Parameters

<i>gnssSvUsedList_len</i>	<ul style="list-style-type: none"> <li>Number of sets of gnssSvUsedList</li> </ul>
<i>pGnssSvUsedList</i>	<ul style="list-style-type: none"> <li>Entry in the list contains the SV ID of a satellite used for calculating this position report.</li> <li>Following information is associated with each SV 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>

## 8.144.2 Field Documentation

8.144.2.1 `uint16_t loc_svUsedforFix::gnssSvUsedList[255]`8.144.2.2 `uint8_t loc_svUsedforFix::gnssSvUsedList_len`8.145 `loc_urlAddr` Struct Reference

## Data Fields

- char [address](#) [256]
- uint8\_t [TlvPresent](#)

### 8.145.1 Detailed Description

This structure contains LOC Get Server URL address field.

#### Parameters

<i>address</i>	<ul style="list-style-type: none"> <li>• Uniform Resource Locator</li> <li>• Type - NULL terminated string</li> <li>• Maximum string length (including NULL terminator) 256</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

### 8.145.2 Field Documentation

8.145.2.1 char loc\_urlAddr::address[256]

8.145.2.2 uint8\_t loc\_urlAddr::TlvPresent

## 8.146 loc\_URLAddrInfo Struct Reference

## Data Fields

- char [urlAddr](#) [256]

### 8.146.1 Detailed Description

This structure specifies information regarding the URL.

#### Parameters

<i>urlAddr</i>	<ul style="list-style-type: none"> <li>• URL address.</li> <li>• Type - NULL-terminated string</li> <li>• Maximum string length (including NULL terminator) - 256</li> </ul>
----------------	--

### 8.146.2 Field Documentation

8.146.2.1 char loc\_URLAddrInfo::urlAddr[256]

## 8.147 IteSSInfo Struct Reference

## Data Fields

- `int8_t` [rssi](#)
- `int8_t` [rsrq](#)
- `int16_t` [rsrp](#)
- `int16_t` [snr](#)

### 8.147.1 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>

### 8.147.2 Field Documentation

8.147.2.1 `int16_t` `lteSSInfo::rsrp`

8.147.2.2 `int8_t` `lteSSInfo::rsrq`

8.147.2.3 `int8_t` `lteSSInfo::rssi`

8.147.2.4 `int16_t` `lteSSInfo::snr`

## 8.148 messageModeTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- [sMSMessageModelInfo](#) [MessageModelInfo](#)

### 8.148.1 Detailed Description

This structure contains message mode TLV information.

## 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>

## 8.148.2 Field Documentation

8.148.2.1 [sMSMessageModelInfo](#) messageModeTlv::MessageModelInfo8.148.2.2 `uint8_t` messageModeTlv::TlvPresent

## 8.149 nas\_acqOrderPref Struct Reference

## Data Fields

- `uint8_t` [acqOrdeLen](#)
- `uint8_t *` [pAcqOrder](#)

## 8.149.1 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>

## 8.149.2 Field Documentation

8.149.2.1 `uint8_t` nas\_acqOrderPref::acqOrdeLen8.149.2.2 `uint8_t*` nas\_acqOrderPref::pAcqOrder

## 8.150 nas\_AcqOrderPrefTlv Struct Reference

## Data Fields

- `uint8_t` [acqOrdeLen](#)

- uint8\_t \* [pAcqOrder](#)
- uint8\_t [TlvPresent](#)

### 8.150.1 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> <li>– 0x0C - NAS_RADIO_IF_NR5G - NR5G</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.150.2 Field Documentation

8.150.2.1 uint8\_t nas\_AcqOrderPrefTlv::acqOrdeLen

8.150.2.2 uint8\_t\* nas\_AcqOrderPrefTlv::pAcqOrder

8.150.2.3 uint8\_t nas\_AcqOrderPrefTlv::TlvPresent

## 8.151 nas\_ActPilotPNElement Struct Reference

### Data Fields

- uint16\_t [ActSetPilotPN](#)
- uint8\_t [ActSetPilotPNStrength](#)

### 8.151.1 Detailed Description

This structure contains pilot PN Element.

#### 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>



### 8.151.2 Field Documentation

8.151.2.1 uint16\_t nas\_ActPilotPNElement::ActSetPilotPN

8.151.2.2 uint8\_t nas\_ActPilotPNElement::ActSetPilotPNStrength

## 8.152 nas\_AddCDMASysInfo Struct Reference

### Data Fields

- uint16\_t [geoSysIdx](#)
- uint16\_t [regPrd](#)

### 8.152.1 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>

### 8.152.2 Field Documentation

8.152.2.1 uint16\_t nas\_AddCDMASysInfo::geoSysIdx

8.152.2.2 uint16\_t nas\_AddCDMASysInfo::regPrd

## 8.153 nas\_AddSysInfo Struct Reference

### Data Fields

- uint16\_t [geoSysIdx](#)
- uint32\_t [cellBroadcastCap](#)

### 8.153.1 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>
--------------------------	--

### 8.153.2 Field Documentation

8.153.2.1 `uint32_t nas_AddSysInfo::cellBroadcastCap`

8.153.2.2 `uint16_t nas_AddSysInfo::geoSysIdx`

## 8.154 nas\_BandPrefInfoTlv Struct Reference

### Data Fields

- `uint64_t` [bits\\_1\\_64](#)
- `uint64_t` [bits\\_65\\_128](#)
- `uint64_t` [bits\\_129\\_192](#)
- `uint64_t` [bits\\_193\\_256](#)
- `uint8_t` [TlvPresent](#)

### 8.154.1 Detailed Description

Contain the Band Preference.

#### Parameters

<i>bits_1_64</i>	<ul style="list-style-type: none"> <li>• Bits 1 to 64 of the 256-bit Operating Band bitmask.</li> </ul>
<i>bits_65_128</i>	<ul style="list-style-type: none"> <li>• Bits 65 to 128 of the 256-bit Operating Band bitmask.</li> </ul>
<i>bits_129_192</i>	<ul style="list-style-type: none"> <li>• Bits 129 to 192 of the 256-bit Operating Band bitmask.</li> </ul>
<i>bits_193_256</i>	<ul style="list-style-type: none"> <li>• Bits 193 to 256 of the 256-bit Operating Band bitmask.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.154.2 Field Documentation

8.154.2.1 `uint64_t nas_BandPrefInfoTlv::bits_129_192`

8.154.2.2 `uint64_t nas_BandPrefInfoTlv::bits_193_256`

8.154.2.3 uint64\_t nas\_BandPrefInfoTlv::bits\_1\_64

8.154.2.4 uint64\_t nas\_BandPrefInfoTlv::bits\_65\_128

8.154.2.5 uint8\_t nas\_BandPrefInfoTlv::TlvPresent

## 8.155 nas\_BandPrefTlv Struct Reference

### Data Fields

- uint64\_t [BandPref](#)
- uint8\_t [TlvPresent](#)

### 8.155.1 Detailed Description

Contain the band preference for system selection preferences.

### Parameters

---

<i>BandPref</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the radio technologies that are disabled.</li> <li>• Bit values:</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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
-------------------	---

## 8.155.2 Field Documentation

8.155.2.1 uint64\_t nas\_BandPrefTlv::BandPref

8.155.2.2 uint8\_t nas\_BandPrefTlv::TlvPresent

## 8.156 nas\_CallBarringSysInfo Struct Reference

### Data Fields

- uint32\_t [csBarStatus](#)
- uint32\_t [psBarStatus](#)

### 8.156.1 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>

## 8.156.2 Field Documentation

8.156.2.1 uint32\_t nas\_CallBarringSysInfo::csBarStatus

8.156.2.2 uint32\_t nas\_CallBarringSysInfo::psBarStatus

## 8.157 nas\_callBarStatus Struct Reference

## Data Fields

- uint32\_t [csBarStatus](#)
- uint32\_t [psBarStatus](#)

### 8.157.1 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>

### 8.157.2 Field Documentation

8.157.2.1 uint32\_t nas\_callBarStatus::csBarStatus

8.157.2.2 uint32\_t nas\_callBarStatus::psBarStatus

## 8.158 nas\_CDMAChannel Struct Reference

## Data Fields

- uint16\_t [priChA](#)
- uint16\_t [priChB](#)
- uint16\_t [secChA](#)
- uint16\_t [secChB](#)

### 8.158.1 Detailed Description

This structure contains CDMA channel.

## 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>

## 8.158.2 Field Documentation

8.158.2.1 uint16\_t nas\_CDMAChannel::priChA

8.158.2.2 uint16\_t nas\_CDMAChannel::priChB

8.158.2.3 uint16\_t nas\_CDMAChannel::secChA

8.158.2.4 uint16\_t nas\_CDMAChannel::secChB

## 8.159 nas\_CDMAECIOThresh Struct Reference

## Data Fields

- uint8\_t [CDMAECIOThreshListLen](#)
- int16\_t \* [pCDMAECIOThreshList](#)

## 8.159.1 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>

## 8.159.2 Field Documentation

8.159.2.1 uint8\_t nas\_CDMAECIOThresh::CDMAECIOThreshListLen

8.159.2.2 `int16_t*` `nas_CDMAECIOThresh::pCDMAECIOThreshList`

## 8.160 `nas_CDMAInfo` Struct Reference

### Data Fields

- `uint16_t` `sid`
- `uint16_t` `nid`
- `uint16_t` `baselId`
- `uint16_t` `refpn`
- `uint32_t` `baseLat`
- `uint32_t` `baseLong`

### 8.160.1 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>baselId</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>

### 8.160.2 Field Documentation

8.160.2.1 `uint16_t` `nas_CDMAInfo::baselId`

8.160.2.2 `uint32_t` `nas_CDMAInfo::baseLat`

8.160.2.3 `uint32_t` `nas_CDMAInfo::baseLong`



8.160.2.4 uint16\_t nas\_CDMAInfo::nid

8.160.2.5 uint16\_t nas\_CDMAInfo::refpn

8.160.2.6 uint16\_t nas\_CDMAInfo::sid

## 8.161 nas\_CDMARSSIThresh Struct Reference

### Data Fields

- uint8\_t [CDMARSSIThreshListLen](#)
- int16\_t \* [pCDMARSSIThreshList](#)

### 8.161.1 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>

### 8.161.2 Field Documentation

8.161.2.1 uint8\_t nas\_CDMARSSIThresh::CDMARSSIThreshListLen

8.161.2.2 int16\_t\* nas\_CDMARSSIThresh::pCDMARSSIThreshList

## 8.162 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]

### 8.162.1 Detailed Description

Structure for storing the CDMA System Information.

#### Parameters

<i>sysInfoCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_sysInfoCommon</a> for more information.</li> </ul>
<i>isSysPrIMatchValid</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[NAS_PLM-N_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[NAS_PLM-N_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>

## 8.162.2 Field Documentation

8.162.2.1 `uint16_t nas_CDMA SysInfo::baseId`

8.162.2.2 `uint32_t nas_CDMA SysInfo::baseLat`

8.162.2.3 `uint32_t nas_CDMA SysInfo::baseLong`

8.162.2.4 `uint8_t nas_CDMA SysInfo::bsInfoValid`

8.162.2.5 `uint8_t nas_CDMA SysInfo::bsPRev`

8.162.2.6 `uint8_t nas_CDMA SysInfo::bsPRevValid`

8.162.2.7 `uint8_t nas_CDMA SysInfo::ccsSupported`

8.162.2.8 `uint8_t nas_CDMA SysInfo::ccsSupportedValid`

8.162.2.9 `uint8_t nas_CDMA SysInfo::cdmaSysIdValid`

8.162.2.10 `uint8_t nas_CDMA SysInfo::isSysPriMatch`

8.162.2.11 `uint8_t nas_CDMA SysInfo::isSysPriMatchValid`

8.162.2.12 `uint8_t nas_CDMA SysInfo::MCC[3]`

8.162.2.13 `uint8_t nas_CDMA SysInfo::MNC[3]`

8.162.2.14 `uint16_t nas_CDMA SysInfo::networkID`

- 8.162.2.15 uint8\_t nas\_CDMA SysInfo::networkIdValid
- 8.162.2.16 uint16\_t nas\_CDMA SysInfo::packetZone
- 8.162.2.17 uint8\_t nas\_CDMA SysInfo::packetZoneValid
- 8.162.2.18 uint8\_t nas\_CDMA SysInfo::pRevInUse
- 8.162.2.19 uint8\_t nas\_CDMA SysInfo::pRevInUseValid
- 8.162.2.20 nas\_sysInfoCommon nas\_CDMA SysInfo::sysInfoCDMA
- 8.162.2.21 uint16\_t nas\_CDMA SysInfo::systemID

## 8.163 nas\_CDMA SysInfoExt Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint8\_t [imsi\\_11\\_12](#)

### 8.163.1 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>

### 8.163.2 Field Documentation

- 8.163.2.1 uint8\_t nas\_CDMA SysInfoExt::imsi\_11\_12
- 8.163.2.2 uint16\_t nas\_CDMA SysInfoExt::MCC

## 8.164 nas\_cellParams Struct Reference

### Data Fields

- uint16\_t [pci](#)
- int16\_t [rsrq](#)
- int16\_t [rsrp](#)
- int16\_t [rssi](#)
- int16\_t [srxlev](#)

### 8.164.1 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>

### 8.164.2 Field Documentation

8.164.2.1 uint16\_t nas\_cellParams::pci

8.164.2.2 int16\_t nas\_cellParams::rsrp

8.164.2.3 int16\_t nas\_cellParams::rsrq

8.164.2.4 int16\_t nas\_cellParams::rssI

8.164.2.5 int16\_t nas\_cellParams::srxlev

## 8.165 nas\_ciotAcqOrderPref Struct Reference

### Data Fields

- uint8\_t [ciotAcqOrderLen](#)
- uint32\_t \* [pCiotAcqOrder](#)

### 8.165.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

## Parameters

<i>ciotAcqOrderLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements.</li> </ul>
<i>pCiotAcqOrder</i>	<ul style="list-style-type: none"> <li>CIOT Acquisition order preference to be set.</li> <li>Values <ul style="list-style-type: none"> <li>0x01 - cdma2000 @1X</li> <li>0x02 - cdma2000 @ HRPD (1xEV-DO)</li> <li>0x04 - GSM</li> <li>0x05 - UMTS</li> <li>0x08 - LTE</li> <li>0x09 - TD-SCDMA</li> <li>0x0a - LTE_M1</li> <li>0x0b - LTE_NB1</li> </ul> </li> </ul>

## 8.165.2 Field Documentation

8.165.2.1 uint8\_t nas\_ciotAcqOrderPref::ciotAcqOrderLen

8.165.2.2 uint32\_t\* nas\_ciotAcqOrderPref::pCiotAcqOrder

## 8.166 nas\_CiotAcqOrderPrefTlv Struct Reference

## Data Fields

- uint8\_t [ciotAcqOrderLen](#)
- uint32\_t \* [pCiotAcqOrder](#)
- uint8\_t [TlvPresent](#)

## 8.166.1 Detailed Description

Contain the CIOT Acquisition Order Preference.

## Parameters

<i>ciotAcqOrderLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements.</li> </ul>
<i>pCiotAcqOrder</i>	<ul style="list-style-type: none"> <li>CIOT Acquisition order preference to be set.</li> <li>Values <ul style="list-style-type: none"> <li>0x01 - cdma2000 @1X</li> <li>0x02 - cdma2000 @ HRPD (1xEV-DO)</li> <li>0x04 - GSM</li> <li>0x05 - UMTS</li> <li>0x08 - LTE</li> <li>0x09 - TD-SCDMA</li> <li>0x0a - LTE_M1</li> <li>0x0b - LTE_NB1</li> </ul> </li> </ul>

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
-------------------	---

### 8.166.2 Field Documentation

8.166.2.1 uint8\_t nas\_CiotAcqOrderPrefTlv::ciotAcqOrderLen

8.166.2.2 uint32\_t\* nas\_CiotAcqOrderPrefTlv::pCiotAcqOrder

8.166.2.3 uint8\_t nas\_CiotAcqOrderPrefTlv::TlvPresent

## 8.167 nas\_CiotLteOpModePrefTlv Struct Reference

### Data Fields

- uint64\_t [ciotLteOpModePref](#)
- uint8\_t [TlvPresent](#)

### 8.167.1 Detailed Description

Contain the CIOT LTE Operational Mode Preference.

#### Parameters

<i>ciotLteOpMode-Pref</i>	<ul style="list-style-type: none"> <li>• CIOT LTE Operational Mode Preference</li> <li>• Values: <ul style="list-style-type: none"> <li>– Bit 0 (0x01) - LTE wideband</li> <li>– Bit 1 (0x02) - LTE M1</li> <li>– Bit 2 (0x04) - LTE NB1 All unlisted bits are reserved for future use and the service point ignores them if used.</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.167.2 Field Documentation

8.167.2.1 uint64\_t nas\_CiotLteOpModePrefTlv::ciotLteOpModePref

8.167.2.2 uint8\_t nas\_CiotLteOpModePrefTlv::TlvPresent

## 8.168 nas\_CommInfo Struct Reference

### Data Fields

- int8\_t [temperature](#)
- uint8\_t [modemMode](#)
- uint8\_t [systemMode](#)
- uint8\_t [imsRegState](#)



- uint8\_t [psState](#)

### 8.168.1 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>
----------------	--

## 8.168.2 Field Documentation

8.168.2.1 `uint8_t nas_CommlInfo::imsRegState`

8.168.2.2 `uint8_t nas_CommlInfo::modemMode`

8.168.2.3 `uint8_t nas_CommlInfo::psState`

8.168.2.4 `uint8_t nas_CommlInfo::systemMode`

8.168.2.5 `int8_t nas_CommlInfo::temperature`

## 8.169 nas\_CsgId Struct Reference

### Data Fields

- `uint32_t csgId`
- `uint8_t TlvPresent`

### 8.169.1 Detailed Description

This structure contains csg ID.

#### Parameters

<i>csgId</i>	<ul style="list-style-type: none"> <li>• Closed subscriber group identifier.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

## 8.169.2 Field Documentation

8.169.2.1 `uint32_t nas_CsgId::csgId`

8.169.2.2 `uint8_t nas_CsgId::TlvPresent`

## 8.170 nas\_CSgid Struct Reference

### Data Fields

- `uint16_t mcc`
- `uint16_t mnc`

- uint8\_t [mncPcsDigits](#)
- uint32\_t [id](#)
- uint8\_t [rat](#)

### 8.170.1 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>

### 8.170.2 Field Documentation

8.170.2.1 uint32\_t nas\_CSGID::id

8.170.2.2 uint16\_t nas\_CSGID::mcc

8.170.2.3 uint16\_t nas\_CSGID::mnc

8.170.2.4 uint8\_t nas\_CSGID::mncPcsDigits

8.170.2.5 uint8\_t nas\_CSGID::rat

## 8.171 nas\_currentPLMN Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [netDescrLength](#)
- uint8\_t [netDescr](#) [255]

### 8.171.1 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>

### 8.171.2 Field Documentation

8.171.2.1 `uint16_t nas_currentPLMN::MCC`

8.171.2.2 `uint16_t nas_currentPLMN::MNC`

8.171.2.3 `uint8_t nas_currentPLMN::netDescr[255]`

8.171.2.4 `uint8_t nas_currentPLMN::netDescrLength`

## 8.172 nas\_dataSrvCapabilities Struct Reference

### Data Fields

- `uint8_t dataCapabilitiesLen`
- `uint8_t dataCapabilities [32]`

### 8.172.1 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>

## 8.172.2 Field Documentation

8.172.2.1 uint8\_t nas\_dataSrvCapabilities::dataCapabilities[32]

8.172.2.2 uint8\_t nas\_dataSrvCapabilities::dataCapabilitiesLen

## 8.173 nas\_DataStatusDetail Struct Reference

## Data Fields

- uint32\_t [IPAddress](#)
- uint8\_t [LastErrCode](#)

## 8.173.1 Detailed Description

This structure contains data status detail

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

## 8.173.2 Field Documentation

8.173.2.1 `uint32_t nas_DataStatusDetail::IPAddress`

8.173.2.2 `uint8_t nas_DataStatusDetail::LastErrCode`

## 8.174 nas\_detailSvcInfo Struct Reference

### Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [srvCapability](#)
- uint8\_t [hdrSrvStatus](#)
- uint8\_t [hdrHybrid](#)
- uint8\_t [isSysForbidden](#)

### 8.174.1 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>
-----------------------	---

## 8.174.2 Field Documentation

8.174.2.1 `uint8_t nas_detailSvcInfo::hdrHybrid`

8.174.2.2 `uint8_t nas_detailSvcInfo::hdrSrvStatus`

8.174.2.3 `uint8_t nas_detailSvcInfo::isSysForbidden`

8.174.2.4 `uint8_t nas_detailSvcInfo::srvCapability`

8.174.2.5 `uint8_t nas_detailSvcInfo::srvStatus`

## 8.175 nas\_DeviceConfigDetail Struct Reference

### Data Fields

- `uint8_t` [Technology](#)
- `uint8_t` [QLIC](#)
- `uint8_t` [Chipset](#)
- `uint8_t` [HWVersion](#)

### 8.175.1 Detailed Description

This structure contains device configure detail.

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

### 8.175.2 Field Documentation

8.175.2.1 `uint8_t nas_DeviceConfigDetail::Chipset`

8.175.2.2 `uint8_t nas_DeviceConfigDetail::HWVersion`

8.175.2.3 `uint8_t nas_DeviceConfigDetail::QLIC`

8.175.2.4 `uint8_t nas_DeviceConfigDetail::Technology`

## 8.176 nas\_dirNum Struct Reference

### Data Fields

- `uint8_t dirNumLen`
- `uint8_t dirNum` [255]

### 8.176.1 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><code>dir_num</code></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>

### 8.176.2 Field Documentation

8.176.2.1 `uint8_t nas_dirNum::dirNum[255]`

8.176.2.2 `uint8_t nas_dirNum::dirNumLen`

## 8.177 nas\_DRCPParams Struct Reference

## Data Fields

- uint8\_t [DRCValue](#)
- uint8\_t [DRCCover](#)

### 8.177.1 Detailed Description

This structure contains DRC information.

#### 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>

### 8.177.2 Field Documentation

8.177.2.1 uint8\_t nas\_DRCParams::DRCCover

8.177.2.2 uint8\_t nas\_DRCParams::DRCValue

## 8.178 nas\_ecioListElement Struct Reference

## Data Fields

- int16\_t [ecio](#)
- uint8\_t [radioIof](#)

### 8.178.1 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>radioIof</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>

### 8.178.2 Field Documentation

8.178.2.1 int16\_t nas\_ecioListElement::ecio

8.178.2.2 uint8\_t nas\_ecioListElement::radiolf

## 8.179 nas\_ECIOThresh Struct Reference

### Data Fields

- uint8\_t [ECIOThresListLen](#)
- int16\_t \* [pECIOThresList](#)

### 8.179.1 Detailed Description

This sturcture contain ECIO threshold list.

#### Parameters

<i>ECIOThresList-Len</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>

### 8.179.2 Field Documentation

8.179.2.1 uint8\_t nas\_ECIOThresh::ECIOThresListLen

8.179.2.2 int16\_t\* nas\_ECIOThresh::pECIOThresList

## 8.180 nas\_EdrxCiotLteMode Struct Reference

### Data Fields

- uint32\_t [lteOpMode](#)
- uint8\_t [TlvPresent](#)

### 8.180.1 Detailed Description

This structure contains the parameters for LTE Operational Mode Indicator Type.

## Parameters

<i>lteOpMode</i>	<ul style="list-style-type: none"> <li>• CIOT LTE mode.</li> <li>• Values <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.180.2 Field Documentation

8.180.2.1 uint32\_t nas\_EdrxCiotLteMode::lteOpMode

8.180.2.2 uint8\_t nas\_EdrxCiotLteMode::TlvPresent

## 8.181 nas\_EdrxCycleLength Struct Reference

## Data Fields

- uint8\_t [cycleLength](#)
- uint8\_t [TlvPresent](#)

## 8.181.1 Detailed Description

This structure contains the parameters for eDRX Cycle Length Indicator Type.

## Parameters

<i>cycleLength</i>	<ul style="list-style-type: none"> <li>• eDRX cycle length. Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.181.2 Field Documentation

8.181.2.1 uint8\_t nas\_EdrxCycleLength::cycleLength

8.181.2.2 uint8\_t nas\_EdrxCycleLength::TlvPresent

## 8.182 nas\_EdrxEnableType Struct Reference

## Data Fields

- uint8\_t [edrxEnabled](#)
- uint8\_t [TlvPresent](#)

### 8.182.1 Detailed Description

This structure contains the parameters for eDRX Enabled Indicator Type.

#### Parameters

<i>edrxEnabled</i>	<ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>0x00 - Disabled</li> <li>0x01 - Enabled</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

### 8.182.2 Field Documentation

8.182.2.1 `uint8_t nas_EdrxEnableType::edrxEnabled`

8.182.2.2 `uint8_t nas_EdrxEnableType::TlvPresent`

## 8.183 nas\_EdrxPagingTimeWindow Struct Reference

#### Data Fields

- `uint8_t edrxPtw`
- `uint8_t TlvPresent`

### 8.183.1 Detailed Description

This structure contains the parameters for eDRX Paging Time Window Indicator Type.

#### Parameters

<i>edrxPtw</i>	<ul style="list-style-type: none"> <li>eDRX paging time window. Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

### 8.183.2 Field Documentation

8.183.2.1 `uint8_t nas_EdrxPagingTimeWindow::edrxPtw`

8.183.2.2 `uint8_t nas_EdrxPagingTimeWindow::TlvPresent`

## 8.184 nas\_EdrxRatType Struct Reference

#### Data Fields

- `uint8_t edrxRatType`
- `uint8_t TlvPresent`

### 8.184.1 Detailed Description

This structure contains the parameters for eDRX RAT type Indicator Type.

#### Parameters

<i>edrxRatType</i>	<ul style="list-style-type: none"> <li>• Radio access technology</li> <li>• Values <ul style="list-style-type: none"> <li>– NAS_RADIO_IF_NO_SVC (0x00) - None (no service)</li> <li>– NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X</li> <li>– NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO)</li> <li>– NAS_RADIO_IF_AMPS (0x03) - AMPS</li> <li>– NAS_RADIO_IF_GSM (0x04) - GSM</li> <li>– NAS_RADIO_IF_UMTS (0x05) - UMTS</li> <li>– NAS_RADIO_IF_WLAN (0x06) - WLAN</li> <li>– NAS_RADIO_IF_GPS (0x07) - GPS</li> <li>– NAS_RADIO_IF_LTE (0x08) - LTE</li> <li>– NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA</li> <li>– NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1</li> <li>– NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1</li> <li>– NAS_RADIO_IF_NO_CHANGE (-1) - No change</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.184.2 Field Documentation

8.184.2.1 `uint8_t nas_EdrxRatType::edrxRatType`

8.184.2.2 `uint8_t nas_EdrxRatType::TlvPresent`

## 8.185 nas\_EmerModeTlv Struct Reference

### Data Fields

- `uint8_t EmerMode`
- `uint8_t TlvPresent`

### 8.185.1 Detailed Description

Contain the emergency Mode parameters for system selection preferences.

#### Parameters

<i>EmerMode</i>	<ul style="list-style-type: none"> <li>• 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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
-------------------	---

## 8.185.2 Field Documentation

8.185.2.1 uint8\_t nas\_EmerModeTlv::EmerMode

8.185.2.2 uint8\_t nas\_EmerModeTlv::TlvPresent

## 8.186 nas\_errorRateListElement Struct Reference

### Data Fields

- uint16\_t [errorRate](#)
- uint8\_t [radiolf](#)

### 8.186.1 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_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>
----------------	---

## 8.186.2 Field Documentation

8.186.2.1 uint16\_t nas\_errorRateListElement::errorRate

8.186.2.2 uint8\_t nas\_errorRateListElement::radioIf

## 8.187 nas\_ForbiddenNetworks3GPP Struct Reference

### Data Fields

- uint16\_t [forbiddenNwInstLen](#)
- uint16\_t [MCC](#) [255]
- uint16\_t [MNC](#) [255]
- uint8\_t [TlvPresent](#)

### 8.187.1 Detailed Description

This structure contains the ForbiddenNetworks3GPP response parameters.

#### Parameters

<i>forbiddenNwInstLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements <ul style="list-style-type: none"> <li>MCC</li> <li>MNC</li> </ul> </li> </ul>
<i>MCC</i>	<ul style="list-style-type: none"> <li>Mobile Country Code</li> <li>A 16-bit integer representation of MCC</li> <li>Range: 0 to 999</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>Mobile Network Code</li> <li>A 16-bit integer representation of MNC</li> <li>Range: 0 to 999</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>



## 8.187.2 Field Documentation

8.187.2.1 uint16\_t nas\_ForbiddenNetworks3GPP::forbiddenNwInstLen

8.187.2.2 uint16\_t nas\_ForbiddenNetworks3GPP::MCC[255]

8.187.2.3 uint16\_t nas\_ForbiddenNetworks3GPP::MNC[255]

8.187.2.4 uint8\_t nas\_ForbiddenNetworks3GPP::TlvPresent

## 8.188 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 [nmrInst](#)
- [nas\\_nmrCellInfo](#) [insNmrCellInfo](#) [255]

### 8.188.1 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_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_nmrCellInfo</a> for more information.</li> </ul>

## 8.188.2 Field Documentation

8.188.2.1 uint16\_t nas\_GERANInfo::arfcn

8.188.2.2 uint8\_t nas\_GERANInfo::bsic

8.188.2.3 uint32\_t nas\_GERANInfo::cellID

8.188.2.4 nas\_nmrCellInfo nas\_GERANInfo::insNmrCellInfo[255]

8.188.2.5 uint16\_t nas\_GERANInfo::lac

8.188.2.6 uint8\_t nas\_GERANInfo::nmlInst

8.188.2.7 uint8\_t nas\_GERANInfo::plmn[3]

8.188.2.8 uint16\_t nas\_GERANInfo::rxLev

8.188.2.9 uint32\_t nas\_GERANInfo::timingAdvance

## 8.189 nas\_geranInstArr Struct Reference

### Data Fields

- uint16\_t [geranArfcn](#)

- uint8\_t [geranBsicNcc](#)
- uint8\_t [geranBsicBcc](#)
- int16\_t [geranRssi](#)
- int16\_t [geranRank](#)

### 8.189.1 Detailed Description

This structure contains information about the GERAN Info.

#### 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>
<i>geranRank</i>	<ul style="list-style-type: none"> <li>• Rank of the cell.</li> </ul>

### 8.189.2 Field Documentation

8.189.2.1 uint16\_t nas\_geranInstArr::geranArfcn

8.189.2.2 uint8\_t nas\_geranInstArr::geranBsicBcc

8.189.2.3 uint8\_t nas\_geranInstArr::geranBsicNcc

8.189.2.4 int16\_t nas\_geranInstArr::geranRank

8.189.2.5 int16\_t nas\_geranInstArr::geranRssi

## 8.190 nas\_geranInstInfo Struct Reference

#### Data Fields

- uint16\_t [geranArfcn](#)
- uint8\_t [geranBsicNcc](#)
- uint8\_t [geranBsicBcc](#)
- int16\_t [geranRssi](#)

### 8.190.1 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>

## 8.190.2 Field Documentation

8.190.2.1 uint16\_t nas\_geranInstInfo::geranArfcn

8.190.2.2 uint8\_t nas\_geranInstInfo::geranBsicBcc

8.190.2.3 uint8\_t nas\_geranInstInfo::geranBsicNcc

8.190.2.4 int16\_t nas\_geranInstInfo::geranRssi

## 8.191 nas\_gsmCellInfo Struct Reference

## Data Fields

- uint16\_t [arfcn](#)
- uint8\_t [band1900](#)
- uint8\_t [cellIdValid](#)
- uint8\_t [bsicId](#)
- int16\_t [rssi](#)
- int16\_t [srxlev](#)

## 8.191.1 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>

## 8.191.2 Field Documentation

8.191.2.1 uint16\_t nas\_gsmCellInfo::arfcn

8.191.2.2 uint8\_t nas\_gsmCellInfo::band1900

8.191.2.3 uint8\_t nas\_gsmCellInfo::bsicId

8.191.2.4 uint8\_t nas\_gsmCellInfo::cellIdValid

8.191.2.5 int16\_t nas\_gsmCellInfo::rsSI

8.191.2.6 int16\_t nas\_gsmCellInfo::srxlev

## 8.192 nas\_GSMRSSIThresh Struct Reference

### Data Fields

- uint8\_t [GSMRSSIThreshListLen](#)
- int16\_t \* [pGSMRSSIThreshList](#)

### 8.192.1 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>

### 8.192.2 Field Documentation

8.192.2.1 `uint8_t nas_GSMRSSIThresh::GSMRSSIThreshListLen`

8.192.2.2 `int16_t* nas_GSMRSSIThresh::pGSMRSSIThreshList`

## 8.193 nas\_GSMSrvStatusInfo Struct Reference

### Data Fields

- `uint8_t srvStatus`
- `uint8_t trueSrvStatus`
- `uint8_t isPrefDataPath`

### 8.193.1 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>

### 8.193.2 Field Documentation

8.193.2.1 `uint8_t nas_GSMSrvStatusInfo::isPrefDataPath`

8.193.2.2 `uint8_t nas_GSMSrvStatusInfo::srvStatus`

8.193.2.3 uint8\_t nas\_GSMsystInfo::trueSrvStatus

## 8.194 nas\_GSMsystInfo Struct Reference

### Data Fields

- [nas\\_systInfoCommon](#) [systInfoGSM](#)
- 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](#)

### 8.194.1 Detailed Description

Structure for storing the GSM System Information.

#### Parameters

<i>systInfoGSM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_systInfoCommon</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 and 10.5.5.14] <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[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[NAS_PLM-N_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>

## 8.194.2 Field Documentation

8.194.2.1 uint32\_t nas\_GSM SysInfo::cellId

8.194.2.2 uint8\_t nas\_GSM SysInfo::cellIdValid

8.194.2.3 uint8\_t nas\_GSM SysInfo::dtmSupp

8.194.2.4 uint8\_t nas\_GSM SysInfo::dtmSuppValid

8.194.2.5 uint8\_t nas\_GSM SysInfo::egprsSupp

8.194.2.6 uint8\_t nas\_GSM SysInfo::egprsSuppValid

8.194.2.7 uint16\_t nas\_GSM SysInfo::lac

8.194.2.8 uint8\_t nas\_GSM SysInfo::lacValid

8.194.2.9 uint8\_t nas\_GSM SysInfo::MCC[3]

8.194.2.10 uint8\_t nas\_GSM SysInfo::MNC[3]

8.194.2.11 uint8\_t nas\_GSM SysInfo::networkIdValid

8.194.2.12 uint8\_t nas\_GSM SysInfo::regRejectInfoValid

8.194.2.13 uint8\_t nas\_GSM SysInfo::rejCause

8.194.2.14 uint8\_t nas\_GSM SysInfo::rejectSrvDomain

8.194.2.15 nas\_sysInfoCommon nas\_GSM SysInfo::sysInfoGSM

## 8.195 nas\_GWAcqOrderPrefTlv Struct Reference

### Data Fields

- uint32\_t [GWAcqOrderPref](#)
- uint8\_t [TlvPresent](#)

### 8.195.1 Detailed Description

Contain the GSM/WCDMA Acquisition order preference for system selection preferences.

#### Parameters

<i>GWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>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>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

### 8.195.2 Field Documentation

8.195.2.1 `uint32_t nas_GWAcqOrderPrefTlv::GWAcqOrderPref`

8.195.2.2 `uint8_t nas_GWAcqOrderPrefTlv::TlvPresent`

## 8.196 nas\_HDRECIOTresh Struct Reference

### Data Fields

- `uint8_t HDRECIOTreshListLen`
- `int16_t * pHDR``ECIOThreshList`

### 8.196.1 Detailed Description

This structure contains HDR ECIO threshold related parameters.

#### Parameters

<i>HDRECIOTreshListLen</i>	<ul style="list-style-type: none"> <li>Length of the HDR ECIO threshold list parameter to follow</li> </ul>
<i>pHDRECIOTreshList</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>

### 8.196.2 Field Documentation

8.196.2.1 `uint8_t nas_HDRECIOTresh::HDRECIOTreshListLen`

8.196.2.2 `int16_t* nas_HDRECIOTresh::pHDRECIOTreshList`

## 8.197 nas\_HDRIOTresh Struct Reference

## Data Fields

- uint8\_t [HDRIOTreshListLen](#)
- int16\_t \* [pHDRIOTreshList](#)

## 8.197.1 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>

## 8.197.2 Field Documentation

8.197.2.1 uint8\_t nas\_HDRIOTresh::HDRIOTreshListLen

8.197.2.2 int16\_t\* nas\_HDRIOTresh::pHDRIOTreshList

## 8.198 nas\_HDRPersonality\_Ind\_Data Struct Reference

## Data Fields

- uint16\_t \* [pCurrentPersonality](#)
- uint8\_t \* [pPersonalityListLength](#)
- [nas\\_protocolSubtypeElement](#) \* [pProtocolSubtypeElement](#)

## 8.198.1 Detailed Description

Elements for HDR Personality indication.

## Parameters

<i>pCurrent- Personality</i>	<ul style="list-style-type: none"> <li>• Current active personality index.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPersonalityList- Length</i>	<ul style="list-style-type: none"> <li>• Number of Personality Protocol Subtype contains in this response.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pProtocol- SubtypeElement</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_protocolSubtypeElement</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

## 8.198.2 Field Documentation

8.198.2.1 `uint16_t* nas_HDRPersonality_Ind_Data::pCurrentPersonality`

8.198.2.2 `uint8_t* nas_HDRPersonality_Ind_Data::pPersonalityListLength`

8.198.2.3 `nas_protocolSubtypeElement* nas_HDRPersonality_Ind_Data::pProtocolSubtypeElement`

## 8.199 nas\_HDRRSSIThresh Struct Reference

### Data Fields

- `uint8_t` [HDRRSSIThreshListLen](#)
- `int16_t *` [pHDRRSSIThreshList](#)

### 8.199.1 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>pHRRSSI- 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>

## 8.199.2 Field Documentation

8.199.2.1 `uint8_t nas_HDRRSSIThresh::HDRRSSIThreshListLen`

8.199.2.2 `int16_t* nas_HDRRSSIThresh::pHDRRSSIThreshList`

## 8.200 nas\_HDRSINRThresh Struct Reference

### Data Fields

- `uint8_t` [HDRSINRThresListLen](#)
- `uint8_t *` [pHDRSINRThresList](#)

### 8.200.1 Detailed Description

This structure contains HDR SINR threshold list.

#### 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- ThreshList</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>
---------------------------------	---

## 8.200.2 Field Documentation

8.200.2.1 `uint8_t nas_HDRSINRThresh::HDRSINRThreshListLen`

8.200.2.2 `uint8_t* nas_HDRSINRThresh::pHDRSINRThreshList`

## 8.201 nas\_HDRSINRThreshold Struct Reference

### Data Fields

- `uint8_t HDRSINRThreshListLen`
- `uint16_t * pHDRSINRThreshList`

### 8.201.1 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>

## 8.201.2 Field Documentation

8.201.2.1 `uint8_t nas_HDRSINRThreshold::HDRSINRThreshListLen`

8.201.2.2 uint16\_t\* nas\_HDRSINRThreshold::pHDRSINRThreshList

## 8.202 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]

### 8.202.1 Detailed Description

Structure for storing the HDR System Information.

#### Parameters

<i>sysInfoHDR</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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>hdrActiveProtValid</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>

## 8.202.2 Field Documentation

8.202.2.1 `uint8_t nas_HDRSysInfo::hdrActiveProt`

8.202.2.2 `uint8_t nas_HDRSysInfo::hdrActiveProtValid`

8.202.2.3 `uint8_t nas_HDRSysInfo::hdrPersonality`

8.202.2.4 `uint8_t nas_HDRSysInfo::hdrPersonalityValid`

8.202.2.5 `uint8_t nas_HDRSysInfo::is856SysId[16]`

8.202.2.6 `uint8_t nas_HDRSysInfo::is856SysIdValid`

8.202.2.7 `uint8_t nas_HDRSysInfo::isSysPrIMatch`

8.202.2.8 `uint8_t nas_HDRSysInfo::isSysPrIMatchValid`

8.202.2.9 nas\_sysInfoCommon nas\_HDRSysInfo::sysInfoHDR

## 8.203 nas\_homeNwMNC3GppTlv Struct Reference

### Data Fields

- uint8\_t [is3GppNw](#)
- uint8\_t [mccIncPcsDigit](#)
- uint8\_t [TlvPresent](#)

### 8.203.1 Detailed Description

This structure contains the 3GPP Home Network MNC parameters.

#### Parameters

<i>is3GppNw[OUT]</i>	<ul style="list-style-type: none"> <li>• Value <ul style="list-style-type: none"> <li>– True - if TLV 0x01 corresponds to a 3GPP network</li> <li>– False - In all other cases.</li> </ul> </li> </ul>
<i>mccIncPcsDigit[-OUT]</i>	<ul style="list-style-type: none"> <li>• This field is used to interpret the length of the mobile_network_code reported in TLV 0x01.</li> <li>• Values <ul style="list-style-type: none"> <li>– TRUE - MNC is a three-digit value if a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a two-digit value if a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>
<i>TlvPresent[OUT]</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

#### Note

mccIncPcsDigit value is ignored if is3GppNw is FALSE.

### 8.203.2 Field Documentation

8.203.2.1 uint8\_t nas\_homeNwMNC3GppTlv::is3GppNw

8.203.2.2 uint8\_t nas\_homeNwMNC3GppTlv::mccIncPcsDigit

8.203.2.3 uint8\_t nas\_homeNwMNC3GppTlv::TlvPresent

## 8.204 nas\_homeSIDNID Struct Reference

### Data Fields

- uint8\_t [numInstances](#)
- [nas\\_sidNid SidNid](#) [255]



### 8.204.1 Detailed Description

This structure contains Home SID/NID

#### 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="#">nas_sidNid</a> for more information</li> </ul>

### 8.204.2 Field Documentation

8.204.2.1 `uint8_t nas_homeSIDNID::numInstances`

8.204.2.2 `nas_sidNid nas_homeSIDNID::SidNid[255]`

## 8.205 nas\_ImVoiceSupportLteTlv Struct Reference

### Data Fields

- `uint8_t ImVoiceSupportLte`
- `uint8_t TLVPresent`

### 8.205.1 Detailed Description

Structure for storing the IMS Voice Support Status on LTE

#### Parameters

<i>ImVoice-SupportLte</i>	<ul style="list-style-type: none"> <li>IMS Voice Support Status on LTE</li> <li>Values <ul style="list-style-type: none"> <li>– 0x00 - Support is not available</li> <li>– 0x01 - Support is available</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>TLV Present</li> </ul>

### 8.205.2 Field Documentation

8.205.2.1 `uint8_t nas_ImVoiceSupportLteTlv::ImVoiceSupportLte`

8.205.2.2 `uint8_t nas_ImVoiceSupportLteTlv::TLVPresent`

## 8.206 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]

### 8.206.1 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 <code>thresh_serving_low</code>, 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 <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 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>

### 8.206.2 Field Documentation

8.206.2.1 uint8\_t `nas_infoInterFreq::cell_resel_priority`

8.206.2.2 `nas_cellParams` `nas_infoInterFreq::cellInterFreqParams`[255]

8.206.2.3 uint8\_t `nas_infoInterFreq::cells_len`

8.206.2.4 uint16\_t `nas_infoInterFreq::earfcn`

8.206.2.5 uint8\_t nas\_infoInterFreq::threshXHigh

8.206.2.6 uint8\_t nas\_infoInterFreq::threshXLow

## 8.207 nas\_IOTresh Struct Reference

### Data Fields

- uint8\_t [IOThresListLen](#)
- int32\_t \* [pIOThresList](#)

### 8.207.1 Detailed Description

This structure contains IO threshold list.

#### Parameters

<i>IOThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the IO 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>

### 8.207.2 Field Documentation

8.207.2.1 uint8\_t nas\_IOTresh::IOThresListLen

8.207.2.2 int32\_t\* nas\_IOTresh::pIOThresList

## 8.208 nas\_IteBandPrefExt Struct Reference

### Data Fields

- uint64\_t [bits\\_1\\_64](#)
- uint64\_t [bits\\_65\\_128](#)
- uint64\_t [bits\\_129\\_192](#)
- uint64\_t [bits\\_193\\_256](#)

### 8.208.1 Detailed Description

Contain the LTE Band Preference Extended parameters

## Parameters

<i>bits_1_64</i>	<ul style="list-style-type: none"> <li>• Bits 1 to 64 of the 256-bit LTE E-UTRA Operating Band bitmask.</li> </ul>
<i>bits_65_128</i>	<ul style="list-style-type: none"> <li>• Bits 65 to 128 of the 256-bit LTE E-UTRA Operating Band bitmask.</li> </ul>
<i>bits_129_192</i>	<ul style="list-style-type: none"> <li>• Bits 129 to 192 of the 256-bit LTE E-UTRA Operating Band bitmask.</li> </ul>
<i>bits_193_256</i>	<ul style="list-style-type: none"> <li>• Bits 193 to 256 of the 256-bit LTE E-UTRA Operating Band bitmask.</li> </ul>

## 8.208.2 Field Documentation

8.208.2.1 uint64\_t nas\_lteBandPrefExt::bits\_129\_192

8.208.2.2 uint64\_t nas\_lteBandPrefExt::bits\_193\_256

8.208.2.3 uint64\_t nas\_lteBandPrefExt::bits\_1\_64

8.208.2.4 uint64\_t nas\_lteBandPrefExt::bits\_65\_128

## 8.209 nas\_LTEBandPrefTlv Struct Reference

## Data Fields

- uint64\_t [LTEBandPref](#)
- uint8\_t [TlvPresent](#)

## 8.209.1 Detailed Description

Contain the LTE Band Preference for system selection preferences.

## Parameters

<i>LTEBandPref</i>	<ul style="list-style-type: none"> <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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.209.2 Field Documentation

8.209.2.1 uint64\_t nas\_LTEBandPrefTlv::LTEBandPref

8.209.2.2 uint8\_t nas\_LTEBandPrefTlv::TlvPresent

## 8.210 nas\_LteCiotOpModeTlv Struct Reference

### Data Fields

- uint32\_t [campedCiotLteOpMode](#)
- uint8\_t [TlvPresent](#)

### 8.210.1 Detailed Description

This structure contains Camped CIOT LTE Operational Mode parameters.

#### Parameters

<i>campedCiotLteOpMode</i>	<ul style="list-style-type: none"> <li>• Indicates the camped CIOT LTE mode of operation.</li> <li>• Values: <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

### 8.210.2 Field Documentation

8.210.2.1 uint32\_t nas\_LteCiotOpModeTlv::campedCiotLteOpMode

8.210.2.2 uint8\_t nas\_LteCiotOpModeTlv::TlvPresent

## 8.211 nas\_LteEARFCN Struct Reference

### Data Fields

- uint8\_t [status](#)
- uint32\_t [earfcn0](#)
- uint32\_t [earfcn1](#)

### 8.211.1 Detailed Description

This structure contains LTE EARFCN information.

#### 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>

### 8.211.2 Field Documentation

8.211.2.1 uint32\_t nas\_LteEARFCN::earfcn0

8.211.2.2 uint32\_t nas\_LteEARFCN::earfcn1

8.211.2.3 uint8\_t nas\_LteEARFCN::status

## 8.212 nas\_LteEarfcnInfo Struct Reference

### Data Fields

- uint8\_t [lteInterEarfcnlen](#)
- uint32\_t [lteEarfcn](#) [255]
- uint8\_t [TlvPresent](#)

### 8.212.1 Detailed Description

This structure contains information about the LTE EARFCN Info.

#### Parameters

<i>lteInterEarfcnlen</i>	<ul style="list-style-type: none"> <li>Number of sets of the lteEarfcn</li> </ul>
<i>lteEarfcn</i>	<ul style="list-style-type: none"> <li>LTE neighbor cell information EARFCN.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

### 8.212.2 Field Documentation

8.212.2.1 uint32\_t nas\_LteEarfcnInfo::lteEarfcn[255]

8.212.2.2 uint8\_t nas\_LteEarfcnInfo::lteInterEarfcnlen

8.212.2.3 uint8\_t nas\_LteEarfcnInfo::TlvPresent

## 8.213 nas\_LteEmbmsCoverageTlv Struct Reference

## Data Fields

- uint8\_t [LteEmbmsCoverage](#)
- uint8\_t [TLVPresent](#)

### 8.213.1 Detailed Description

Structure for storing the LTE eMBMS Coverage Information.

#### Parameters

<i>LteEmbms-Coverage</i>	<ul style="list-style-type: none"> <li>• Values <ul style="list-style-type: none"> <li>– TRUE - Current LTE system supports eMBMS</li> <li>– FALSE - Current LTE system does not support eMBMS</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.213.2 Field Documentation

8.213.2.1 uint8\_t nas\_LteEmbmsCoverageTlv::LteEmbmsCoverage

8.213.2.2 uint8\_t nas\_LteEmbmsCoverageTlv::TLVPresent

## 8.214 nas\_LteEmbmsTraceIdTlv Struct Reference

## Data Fields

- int16\_t [LteEmbmsTraceId](#)
- uint8\_t [TLVPresent](#)

### 8.214.1 Detailed Description

Structure for storing the LTE eMBMS Coverage Info Trace ID.

#### Parameters

<i>LteEmbmsTrace-Id</i>	<ul style="list-style-type: none"> <li>• LTE eMBMS coverage information trace ID.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 to 32768 - Valid trace ID</li> <li>– -1 - Trace ID is not used</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.214.2 Field Documentation

8.214.2.1 int16\_t nas\_LteEmbmsTraceIdTlv::LteEmbmsTraceId



8.214.2.2 uint8\_t nas\_LteEmbmsTraceldTlv::TLVPresent

## 8.215 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]

### 8.215.1 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>

### 8.215.2 Field Documentation

8.215.2.1 uint8\_t nas\_LteGsmCellInfo::cellReselPriority

8.215.2.2 `uint8_t nas_lteGsmCellInfo::cells_len`

8.215.2.3 `nas_gsmCellInfo nas_lteGsmCellInfo::GsmCellInfo[255]`

8.215.2.4 `uint8_t nas_lteGsmCellInfo::nccPermitted`

8.215.2.5 `uint8_t nas_lteGsmCellInfo::threshGsmHigh`

8.215.2.6 `uint8_t nas_lteGsmCellInfo::threshGsmLow`

## 8.216 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`

### 8.216.1 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>

## 8.216.2 Field Documentation

### 8.216.2.1 uint8\_t nas\_LTEInfo::band

8.216.2.2 `uint8_t nas_LTEInfo::bandwidth`

8.216.2.3 `uint8_t nas_LTEInfo::emmConnState`

8.216.2.4 `uint8_t nas_LTEInfo::emmState`

8.216.2.5 `uint8_t nas_LTEInfo::emmSubState`

8.216.2.6 `uint16_t nas_LTEInfo::RXChan`

8.216.2.7 `uint16_t nas_LTEInfo::TXChan`

## 8.217 nas\_LTEInfoInterfreq Struct Reference

### Data Fields

- `uint8_t uelIdle`
- `uint8_t freqsLen`
- `nas_infoInterFreq InfoInterfreq` [255]

### 8.217.1 Detailed Description

This structure contains information about the LTE Inter-Frequency Network.

#### Parameters

<i>uelIdle</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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_infoInterFreq</a> for more information.</li> </ul>

### 8.217.2 Field Documentation

8.217.2.1 `uint8_t nas_LTEInfoInterfreq::freqsLen`

8.217.2.2 `nas_infoInterFreq nas_LTEInfoInterfreq::InfoInterfreq`[255]

8.217.2.3 `uint8_t nas_LTEInfoInterfreq::uelIdle`

## 8.218 nas\_LTEInfoIntrafreq Struct Reference

### Data Fields

- `uint8_t uelIdle`
- `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]

### 8.218.1 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 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_cellParams</a> for more information.</li> </ul>

## 8.218.2 Field Documentation

8.218.2.1 `nas_cellParams nas_LTEInfoIntrafreq::CellParams[255]`

8.218.2.2 `uint8_t nas_LTEInfoIntrafreq::cellReselPriority`

8.218.2.3 `uint8_t nas_LTEInfoIntrafreq::cellsLen`

8.218.2.4 `uint16_t nas_LTEInfoIntrafreq::earfcn`

8.218.2.5 `uint32_t nas_LTEInfoIntrafreq::globalCellId`

8.218.2.6 `uint8_t nas_LTEInfoIntrafreq::plmn[3]`

8.218.2.7 `uint16_t nas_LTEInfoIntrafreq::servingCellId`

8.218.2.8 `uint8_t nas_LTEInfoIntrafreq::sIntraSearch`

8.218.2.9 `uint8_t nas_LTEInfoIntrafreq::sNonIntraSearch`

8.218.2.10 `uint16_t nas_LTEInfoIntrafreq::tac`

8.218.2.11 `uint8_t nas_LTEInfoIntrafreq::threshServingLow`

8.218.2.12 `uint8_t nas_LTEInfoIntrafreq::ueInIdle`

## 8.219 nas\_LTEInfoNeighboringGSM Struct Reference

## Data Fields

- uint8\_t [ueInIdle](#)
- uint8\_t [freqsLen](#)
- [nas\\_LteGsmCellInfo](#) [LteGsmCellInfo](#) [255]

### 8.219.1 Detailed Description

This structure contains information about the LTE Neighboring GSM 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 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>

### 8.219.2 Field Documentation

8.219.2.1 uint8\_t nas\_LTEInfoNeighboringGSM::freqsLen

8.219.2.2 [nas\\_LteGsmCellInfo](#) nas\_LTEInfoNeighboringGSM::LteGsmCellInfo[255]

8.219.2.3 uint8\_t nas\_LTEInfoNeighboringGSM::ueInIdle

## 8.220 nas\_LTEInfoNeighboringWCDMA Struct Reference

## Data Fields

- uint8\_t [ueInIdle](#)
- uint8\_t [freqsLen](#)
- [nas\\_LteWcdmaCellInfo](#) [LTEWCDMACellInfo](#) [255]

### 8.220.1 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[NAS_M-AX_DESCRIPTOR_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_lteWcdmaCellInfo</a> for more information.</li> </ul>

## 8.220.2 Field Documentation

8.220.2.1 `uint8_t nas_LTEInfoNeighboringWCDMA::freqsLen`

8.220.2.2 `nas_lteWcdmaCellInfo nas_LTEInfoNeighboringWCDMA::LTEWCDMACellInfo[255]`

8.220.2.3 `uint8_t nas_LTEInfoNeighboringWCDMA::uelIdle`

## 8.221 nas\_LteM1BandPrefTlv Struct Reference

### Data Fields

- `uint64_t lteM1BandPref`
- `uint8_t TlvPresent`

### 8.221.1 Detailed Description

Contain the LTE M1 band Preference.

#### Parameters

---



<i>lteM1BandPref</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE M1 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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.221.2 Field Documentation

8.221.2.1    `uint64_t nas_LteM1BandPrefTlv::lteM1BandPref`

8.221.2.2    `uint8_t nas_LteM1BandPrefTlv::TlvPresent`

## 8.222    `nas_LteNb1BandPrefTlv` Struct Reference

### Data Fields

- `uint64_t` [lteNb1BandPref](#)
- `uint8_t` [TlvPresent](#)

### 8.222.1    Detailed Description

Contain the LTE NB1 band Preference.

### Parameters

---

<i>LteNb1BandPref</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE NB1 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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.222.2 Field Documentation

8.222.2.1 uint64\_t nas\_LteNb1BandPrefTlv::lteNb1BandPref

8.222.2.2 uint8\_t nas\_LteNb1BandPrefTlv::TlvPresent

## 8.223 nas\_LTEOperationalModeTlv Struct Reference

### Data Fields

- uint32\_t [ciotLteOpMode](#)
- uint8\_t [TlvPresent](#)

### 8.223.1 Detailed Description

Contain fields in struct [nas\\_LTEOperationalModeTlv](#)

#### Parameters

<i>ciotLteOpMode</i>	<ul style="list-style-type: none"> <li>• CIOT LTE operational mode.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Camped on LTE wideband</li> <li>– 0x02 - Camped on LTE M1</li> <li>– 0x03 - Camped on LTE NB1</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

#### Note

None

### 8.223.2 Field Documentation

8.223.2.1 uint32\_t nas\_LTEOperationalModeTlv::ciotLteOpMode

8.223.2.2 uint8\_t nas\_LTEOperationalModeTlv::TlvPresent

## 8.224 nas\_LTEOperationMode Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t \* [pLTEOperationMode](#)

### 8.224.1 Detailed Description

This structure contains the [nas\\_LTEOperationMode](#) response parameters.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<i>pLTEOperation-Mode</i>	<ul style="list-style-type: none"> <li>• LTE Operational Mode</li> </ul>

## 8.224.2 Field Documentation

8.224.2.1 uint32\_t\* nas\_LTEOperationMode::pLTEOperationMode

8.224.2.2 uint8\_t nas\_LTEOperationMode::TlvPresent

## 8.225 nas\_LteOpMode Struct Reference

## Data Fields

- uint32\_t [lteOpMode](#)
- uint8\_t [TlvPresent](#)

## 8.225.1 Detailed Description

This structure contains LTE Operational Mode

## Parameters

<i>lteOpMode</i>	<ul style="list-style-type: none"> <li>• LTE Operational Mode</li> <li>• CIOT LTE mode on which reject indication is received.</li> <li>• Values <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

## 8.225.2 Field Documentation

8.225.2.1 uint32\_t nas\_LteOpMode::lteOpMode

8.225.2.2 uint8\_t nas\_LteOpMode::TlvPresent

## 8.226 nas\_IteOpModeTlv Struct Reference

## Data Fields

- uint8\_t [lteOpModeLen](#)
- uint16\_t [MCC](#) [255]

- uint16\_t [MNC](#) [255]
- uint32\_t [lteOpMode](#) [255]
- uint8\_t [TlvPresent](#)

### 8.226.1 Detailed Description

This structure contains the LTE Operational Mode response parameters.

#### Parameters

<i>lteOpModeLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– MCC</li> <li>– MNC</li> <li>– lteOpMode</li> </ul> </li> </ul>
<i>MCC</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code</li> <li>• A 16-bit integer representation of MCC</li> <li>• Range: 0 to 999</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>• Mobile Network Code</li> <li>• A 16-bit integer representation of MNC</li> <li>• Range: 0 to 999</li> </ul>
<i>lteOpMode</i>	<ul style="list-style-type: none"> <li>• Indicates the LTE mode of operation.</li> <li>• Values: <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.226.2 Field Documentation

8.226.2.1 uint32\_t nas\_lteOpModeTlv::lteOpMode[255]

8.226.2.2 uint8\_t nas\_lteOpModeTlv::lteOpModeLen

8.226.2.3 uint16\_t nas\_lteOpModeTlv::MCC[255]

8.226.2.4 uint16\_t nas\_lteOpModeTlv::MNC[255]

8.226.2.5 uint8\_t nas\_lteOpModeTlv::TlvPresent

### 8.227 nas\_ltePCI Struct Reference

## Data Fields

- uint8\_t [status](#)
- uint32\_t [earfcn](#)
- uint32\_t [pci](#)

### 8.227.1 Detailed Description

This structure contains LTE PCI information.

#### 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>

### 8.227.2 Field Documentation

8.227.2.1 uint32\_t nas\_ltePCI::earfcn

8.227.2.2 uint32\_t nas\_ltePCI::pci

8.227.2.3 uint8\_t nas\_ltePCI::status

## 8.228 nas\_LteRegDomainTlv Struct Reference

## Data Fields

- uint32\_t [LteRegDomain](#)
- uint8\_t [TLVPresent](#)

### 8.228.1 Detailed Description

Structure for storing the LTE Registration Domain

#### Parameters

<i>LteRegDomain</i>	<ul style="list-style-type: none"> <li>• LTE registration domain.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Not applicable because the UE is not in Camp Only mode</li> <li>– 1 - UE is in Camp Only mode and the PLMN can provide CS service only</li> <li>– 2 - UE is in Camp Only mode and the PLMN can provide PS service only</li> <li>– 3 - UE is in Camp Only mode and the PLMN can provide CS and PS service</li> <li>– 4 - UE is in Camp Only</li> </ul> </li> </ul>
---------------------	--

<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>
-------------------	---

### 8.228.2 Field Documentation

8.228.2.1 uint32\_t nas\_LteRegDomainTlv::LteRegDomain

8.228.2.2 uint8\_t nas\_LteRegDomainTlv::TLVPresent

## 8.229 nas\_LteRsrpinformation Struct Reference

### Data Fields

- int16\_t [rsrplevel](#)

### 8.229.1 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>
------------------	--

### 8.229.2 Field Documentation

8.229.2.1 int16\_t nas\_LteRsrpinformation::rsrplevel

## 8.230 nas\_LTERSRPThresh Struct Reference

### Data Fields

- uint8\_t [LTERSRPThreshListLen](#)
- int16\_t \* [pLTERSRPThreshList](#)

### 8.230.1 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>



### 8.230.2 Field Documentation

8.230.2.1 uint8\_t nas\_LTERSRPThresh::LTERSRPThreshListLen

8.230.2.2 int16\_t\* nas\_LTERSRPThresh::pLTERSRPThreshList

## 8.231 nas\_LTERSRQThresh Struct Reference

### Data Fields

- uint8\_t [LTERSRQThreshListLen](#)
- int16\_t \* [pLTERSRQThreshList](#)

### 8.231.1 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>

### 8.231.2 Field Documentation

8.231.2.1 uint8\_t nas\_LTERSRQThresh::LTERSRQThreshListLen

8.231.2.2 int16\_t\* nas\_LTERSRQThresh::pLTERSRQThreshList

## 8.232 nas\_LTERSSIThresh Struct Reference

### Data Fields

- uint8\_t [LTERSSIThreshListLen](#)
- int16\_t \* [pLTERSSIThreshList](#)

### 8.232.1 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>

## 8.232.2 Field Documentation

8.232.2.1 `uint8_t nas_LTERSSIThresh::LTERSSIThreshListLen`8.232.2.2 `int16_t* nas_LTERSSIThresh::pLTERSSIThreshList`8.233 `nas_LTESigRptCfg` Struct Reference

## Data Fields

- `uint8_t rptRate`
- `uint8_t avgPeriod`

## 8.233.1 Detailed Description

This structure contains LTE signal report config.

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

### 8.233.2 Field Documentation

8.233.2.1 uint8\_t nas\_LTESigRptCfg::avgPeriod

8.233.2.2 uint8\_t nas\_LTESigRptCfg::rptRate

## 8.234 nas\_LTESigRptConfig Struct Reference

### Data Fields

- uint8\_t [rptRate](#)
- uint8\_t [avgPeriod](#)

### 8.234.1 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>
------------------	--

### 8.234.2 Field Documentation

8.234.2.1 uint8\_t nas\_LTESigRptConfig::avgPeriod

8.234.2.2 uint8\_t nas\_LTESigRptConfig::rptRate

## 8.235 nas\_IteSnrinformation Struct Reference

### Data Fields

- int16\_t [snrlevel](#)

### 8.235.1 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>
-----------------	---

### 8.235.2 Field Documentation

8.235.2.1 int16\_t nas\_IteSnrinformation::snrlevel

## 8.236 nas\_LTESNRThresh Struct Reference

### Data Fields

- uint8\_t [LTESNRThresListLen](#)
- int16\_t \* [pLTESNRThresList](#)

### 8.236.1 Detailed Description

This structure contains LTE SNR threshold list.

#### 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>

### 8.236.2 Field Documentation

8.236.2.1 `uint8_t nas_LTESNRThresh::LTESNRThresListLen`

8.236.2.2 `int16_t* nas_LTESNRThresh::pLTESNRThresList`

## 8.237 nas\_LTESNRThreshold Struct Reference

### Data Fields

- `uint8_t` [LTESNRThreshListLen](#)
- `int16_t *` [pLTESNRThreshList](#)

### 8.237.1 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>

### 8.237.2 Field Documentation

8.237.2.1 `uint8_t nas_LTESNRThreshold::LTESNRThreshListLen`

8.237.2.2 int16\_t\* nas\_LTESNRThreshold::pLTESNRThreshList

## 8.238 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](#)

### 8.238.1 Detailed Description

Structure for storing the LTE System Information.

#### Parameters

<i>sysInfoLTE</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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]. and [3GPP TS 24.301, Section 9.9.3.9] <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[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> <li>Mobile Country Code.</li> <li>MCC digits in ASCII characters</li> </ul>
<i>MNC[NAS_PLM-N_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>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>

## 8.238.2 Field Documentation

- 8.238.2.1 uint32\_t nas\_LTESysInfo::cellId
- 8.238.2.2 uint8\_t nas\_LTESysInfo::cellIdValid
- 8.238.2.3 uint16\_t nas\_LTESysInfo::lac
- 8.238.2.4 uint8\_t nas\_LTESysInfo::lacValid
- 8.238.2.5 uint8\_t nas\_LTESysInfo::MCC[3]
- 8.238.2.6 uint8\_t nas\_LTESysInfo::MNC[3]
- 8.238.2.7 uint8\_t nas\_LTESysInfo::networkIdValid
- 8.238.2.8 uint8\_t nas\_LTESysInfo::regRejectInfoValid
- 8.238.2.9 uint8\_t nas\_LTESysInfo::rejCause
- 8.238.2.10 uint8\_t nas\_LTESysInfo::rejectSrvDomain
- 8.238.2.11 nas\_sysInfoCommon nas\_LTESysInfo::sysInfoLTE
- 8.238.2.12 uint16\_t nas\_LTESysInfo::tac
- 8.238.2.13 uint8\_t nas\_LTESysInfo::tacValid

## 8.239 nas\_LteVoiceDomainTlv Struct Reference

### Data Fields

- uint32\_t [LteVoiceDomain](#)
- uint8\_t [TLVPresent](#)

### 8.239.1 Detailed Description

Structure for storing the LTE Voice Domain

#### Parameters

<i>LteVoiceDomain</i>	<ul style="list-style-type: none"> <li>• LTE voice domain.             <ul style="list-style-type: none"> <li>– Values                 <ul style="list-style-type: none"> <li>* 0 - Data-centric devices: No voice, stay on LTE</li> <li>* 1 - Voice is supported over the IMS network</li> <li>* 2 - Voice is supported over the 1X network</li> <li>* 3 - Voice is supported over the 3GPP network</li> </ul> </li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.239.2 Field Documentation



8.239.2.1 uint32\_t nas\_LteVoiceDomainTlv::LteVoiceDomain

8.239.2.2 uint8\_t nas\_LteVoiceDomainTlv::TLVPresent

## 8.240 nas\_IteWcdmaCellInfo Struct Reference

### Data Fields

- uint16\_t [uarfcn](#)
- uint8\_t [cellReselPriority](#)
- uint16\_t [threshXhigh](#)
- uint16\_t [threshXlow](#)
- uint8\_t [cellsLen](#)
- [nas\\_wcdmaCellInfo WCDMACellInfo](#) [255]

### 8.240.1 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[NAS_MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_wcdmaCellInfo</a> for more information.</li> </ul>

### 8.240.2 Field Documentation

8.240.2.1 uint8\_t nas\_IteWcdmaCellInfo::cellReselPriority

8.240.2.2 uint8\_t nas\_IteWcdmaCellInfo::cellsLen

8.240.2.3 uint16\_t nas\_lteWcdmaCellInfo::threshXhigh

8.240.2.4 uint16\_t nas\_lteWcdmaCellInfo::threshXlow

8.240.2.5 uint16\_t nas\_lteWcdmaCellInfo::uarfcn

8.240.2.6 nas\_wcdmaCellInfo nas\_lteWcdmaCellInfo::WCDMACellInfo[255]

## 8.241 nas\_Mdn Struct Reference

### Data Fields

- uint8\_t [mdnLen](#)
- uint8\_t [mdn](#) [15]

### 8.241.1 Detailed Description

This structure contains the parameters for CDMA MDN Information

#### Parameters

<i>mdnLen</i>	<ul style="list-style-type: none"> <li>• Length of MDN</li> </ul>
<i>MDN</i>	<ul style="list-style-type: none"> <li>• MDN in ASCII format, max length is 15</li> </ul>

### 8.241.2 Field Documentation

8.241.2.1 uint8\_t nas\_Mdn::mdn[15]

8.241.2.2 uint8\_t nas\_Mdn::mdnLen

## 8.242 nas\_minBasedIMSI Struct Reference

### Data Fields

- uint8\_t [mccM](#) [3]
- uint16\_t [imsiM1112](#)
- uint8\_t [imsiMS1](#) [7]
- uint8\_t [imsiMS2](#) [3]

### 8.242.1 Detailed Description

This structure contains MIN-based IMSI.

#### 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>

## 8.242.2 Field Documentation

8.242.2.1 uint16\_t nas\_minBasedIMSI::imsiM1112

8.242.2.2 uint8\_t nas\_minBasedIMSI::imsiMS1[7]

8.242.2.3 uint8\_t nas\_minBasedIMSI::imsiMS2[3]

8.242.2.4 uint8\_t nas\_minBasedIMSI::mccM[3]

## 8.243 nas\_MNCPCSDigitStatus Struct Reference

### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [MNCIncPCSDigit](#)
- uint8\_t [TLVPresent](#)

### 8.243.1 Detailed Description

This structure contains MNC PCS Digit Include Status.

#### Parameters

<i>MCC</i>	<ul style="list-style-type: none"> <li>• A 16-bit integer representation of MCC.</li> <li>• Range - 0 to 999.</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>• A 16-bit integer representation of MNC.</li> <li>• Range - 0 to 999.</li> </ul>
<i>MNCIncPCS-Digit</i>	<ul style="list-style-type: none"> <li>• This field is used to interpret the length of the corresponding MNC reported in the TLVs with an MNC.</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>

<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present.</li> </ul>
-------------------	--

### 8.243.2 Field Documentation

8.243.2.1 uint16\_t nas\_MNCPCSDigitStatus::MCC

8.243.2.2 uint16\_t nas\_MNCPCSDigitStatus::MNC

8.243.2.3 uint8\_t nas\_MNCPCSDigitStatus::MNCIncPCSDigit

8.243.2.4 uint8\_t nas\_MNCPCSDigitStatus::TLVPresent

## 8.244 nas\_MNRInfo Struct Reference

### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- uint32\_t [rat](#)

### 8.244.1 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>

### 8.244.2 Field Documentation

8.244.2.1 uint16\_t nas\_MNRInfo::mcc

8.244.2.2 uint16\_t nas\_MNRInfo::mnc

8.244.2.3 uint32\_t nas\_MNRInfo::rat

## 8.245 nas\_ModePrefTlv Struct Reference

## Data Fields

- uint16\_t [ModePref](#)
- uint8\_t [TlvPresent](#)

## 8.245.1 Detailed Description

Contain the radio technology mode preference

## Parameters

<i>ModePref</i>	<ul style="list-style-type: none"> <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> <li>– Bit 5 - TD-SCDMA</li> <li>– Bit 6 - NR5G</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.245.2 Field Documentation

8.245.2.1 uint16\_t nas\_ModePrefTlv::ModePref

8.245.2.2 uint8\_t nas\_ModePrefTlv::TlvPresent

## 8.246 nas\_namName Struct Reference

## Data Fields

- uint8\_t [namNameLen](#)
- uint8\_t [namName](#) [12]

## 8.246.1 Detailed Description

This structure contains NAM Name.

## 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>– nam_name</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 nam_name is 12.</li> </ul>

## 8.246.2 Field Documentation

8.246.2.1 `uint8_t nas_namName::namName[12]`

8.246.2.2 `uint8_t nas_namName::namNameLen`

## 8.247 nas\_netSelectionPref Struct Reference

### Data Fields

- `uint8_t netReg`
- `uint16_t mcc`
- `uint16_t mnc`

### 8.247.1 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>

## 8.247.2 Field Documentation

8.247.2.1 `uint16_t nas_netSelectionPref::mcc`

8.247.2.2 `uint16_t nas_netSelectionPref::mnc`

8.247.2.3 `uint8_t nas_netSelectionPref::netReg`

## 8.248 nas\_NetSelPrefTlv Struct Reference

### Data Fields

- `uint8_t NetSelPref`
- `uint8_t TlvPresent`

### 8.248.1 Detailed Description

Contain the Network selection Preference for system selection preferences.

## Parameters

<i>NetSelPref</i>	<ul style="list-style-type: none"> <li>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>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

## 8.248.2 Field Documentation

8.248.2.1 uint8\_t nas\_NetSelPrefTlv::NetSelPref

8.248.2.2 uint8\_t nas\_NetSelPrefTlv::TlvPresent

## 8.249 nas\_networkNameSrcTlv Struct Reference

## Data Fields

- uint8\_t [nwNameSrcLen](#)
- uint32\_t [nwNameSrc](#) [255]
- uint8\_t [TlvPresent](#)

## 8.249.1 Detailed Description

This structure contains the Network Name Source parameters.

## Parameters

<i>nwNameSrcLen</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements <ul style="list-style-type: none"> <li>nwNameSrc</li> </ul> </li> </ul>
<i>nwNameSrc</i>	<ul style="list-style-type: none"> <li>Network name source.</li> <li>Values <ul style="list-style-type: none"> <li>0x00 - Unknown</li> <li>0x01 - Operator PLMN list and PLMN network name</li> <li>0x02 - Common PCN handset specification and operator name string</li> <li>0x03 - Network identity and time zone</li> <li>0x04 - GSMA SE13 table</li> <li>0x05 - Mobile country code and mobile network code</li> <li>0x06 - Service provider name</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

## 8.249.2 Field Documentation

8.249.2.1 uint32\_t nas\_networkNameSrcTlv::nwNameSrc[255]

8.249.2.2 uint8\_t nas\_networkNameSrcTlv::nwNameSrcLen

8.249.2.3 uint8\_t nas\_networkNameSrcTlv::TlvPresent

## 8.250 nas\_NetworkStat1x Struct Reference

### Data Fields

- uint8\_t [State](#)
- uint16\_t [SO](#)
- uint32\_t [RX\\_PWR](#)
- uint16\_t [RX\\_EC\\_IO](#)
- uint32\_t [TX\\_PWR](#)
- uint8\_t [ActSetCnt](#)
- [nas\\_ActPilotPNElement](#) \* [pActPilotPNElements](#)
- uint8\_t [NeighborSetCnt](#)
- uint16\_t \* [pNeighborSetPilotPN](#)

### 8.250.1 Detailed Description

This structure contains network stat 1x.

#### 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>	<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="#">nas_ActPilotPNElement</a> for more information</li> </ul>
<i>NeighborSetCnt</i>	<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>

## 8.250.2 Field Documentation

8.250.2.1 uint8\_t nas\_NetworkStat1x::ActSetCnt

8.250.2.2 uint8\_t nas\_NetworkStat1x::NeighborSetCnt

8.250.2.3 nas\_ActPilotPNElement\* nas\_NetworkStat1x::pActPilotPNElements

8.250.2.4 uint16\_t\* nas\_NetworkStat1x::pNeighborSetPilotPN

8.250.2.5 uint16\_t nas\_NetworkStat1x::RX\_EC\_IO

8.250.2.6 uint32\_t nas\_NetworkStat1x::RX\_PWR

8.250.2.7 uint16\_t nas\_NetworkStat1x::SO

8.250.2.8 uint8\_t nas\_NetworkStat1x::State

8.250.2.9 uint32\_t nas\_NetworkStat1x::TX\_PWR

## 8.251 nas\_NetworkStatEVDO Struct Reference

### Data Fields

- uint8\_t [State](#)
- uint8\_t [MACIndex](#)
- uint8\_t [SectorIDLen](#)
- uint16\_t \* [pSectorID](#)
- uint16\_t [RX\\_PWR](#)
- uint16\_t [PER](#)
- uint16\_t [PilotEnergy](#)
- uint8\_t [SNR](#)

### 8.251.1 Detailed Description

This structure contains network stat EVDO.

#### 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>

## 8.251.2 Field Documentation

8.251.2.1 uint8\_t nas\_NetworkStatEVDO::MACIndex

8.251.2.2 uint16\_t nas\_NetworkStatEVDO::PER

8.251.2.3 uint16\_t nas\_NetworkStatEVDO::PilotEnergy

8.251.2.4 uint16\_t\* nas\_NetworkStatEVDO::pSectorID

8.251.2.5 uint16\_t nas\_NetworkStatEVDO::RX\_PWR

8.251.2.6 uint8\_t nas\_NetworkStatEVDO::SectorIDLen

8.251.2.7 uint8\_t nas\_NetworkStatEVDO::SNR

8.251.2.8 uint8\_t nas\_NetworkStatEVDO::State

## 8.252 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](#)

### 8.252.1 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>

### 8.252.2 Field Documentation

8.252.2.1 uint16\_t nas\_nmrCellInfo::nrmArfcn

8.252.2.2 uint8\_t nas\_nmrCellInfo::nrmBsic

8.252.2.3 uint32\_t nas\_nmrCellInfo::nrmCellID

8.252.2.4 uint16\_t nas\_nmrCellInfo::nrmLac

8.252.2.5 uint8\_t nas\_nmrCellInfo::nmrPlmn[3]

8.252.2.6 uint16\_t nas\_nmrCellInfo::nmrRxLev

## 8.253 nas\_nr5gBandPref Struct Reference

### Data Fields

- uint64\_t [bits\\_1\\_64](#)
- uint64\_t [bits\\_65\\_128](#)
- uint64\_t [bits\\_129\\_192](#)
- uint64\_t [bits\\_193\\_256](#)

### 8.253.1 Detailed Description

Contain the NR5G Band Preference.

#### Parameters

<i>bits_1_64</i>	<ul style="list-style-type: none"> <li>• Bits 1 to 64 of the 256-bit NR5G Operating Band bitmask.</li> </ul>
<i>bits_65_128</i>	<ul style="list-style-type: none"> <li>• Bits 65 to 128 of the 256-bit NR5G Operating Band bitmask.</li> </ul>
<i>bits_129_192</i>	<ul style="list-style-type: none"> <li>• Bits 129 to 192 of the 256-bit NR5G Operating Band bitmask.</li> </ul>
<i>bits_193_256</i>	<ul style="list-style-type: none"> <li>• Bits 193 to 256 of the 256-bit NR5G Operating Band bitmask.</li> </ul>

### 8.253.2 Field Documentation

8.253.2.1 uint64\_t nas\_nr5gBandPref::bits\_129\_192

8.253.2.2 uint64\_t nas\_nr5gBandPref::bits\_193\_256

8.253.2.3 uint64\_t nas\_nr5gBandPref::bits\_1\_64

8.253.2.4 uint64\_t nas\_nr5gBandPref::bits\_65\_128

## 8.254 nas\_NR5GCellStatusInfoTlv Struct Reference

### Data Fields

- uint32\_t [nr5gCellStatus](#)
- uint8\_t [TLVPresent](#)

### 8.254.1 Detailed Description

Structure for storing the NR5G Cell Access Status Info

## Parameters

<i>nr5gCellStatus</i>	<ul style="list-style-type: none"> <li>Cell access status for NR5G calls. <ul style="list-style-type: none"> <li>Values</li> <li>0x00 - Cell access is allowed for normal calls only</li> <li>0x01 - Cell access is allowed for emergency calls only</li> <li>0x02 - Cell access is not allowed for any call type</li> <li>0x03 - Cell access is allowed for all call types</li> <li>-1 - Cell access type is unknown</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>TLV Present</li> </ul>

## 8.254.2 Field Documentation

8.254.2.1 uint32\_t nas\_NR5GCellStatusInfoTlv::nr5gCellStatus

8.254.2.2 uint8\_t nas\_NR5GCellStatusInfoTlv::TLVPresent

## 8.255 nas\_NR5GCellStatusTlv Struct Reference

## Data Fields

- uint32\_t [nr5gCellStatus](#)
- uint8\_t [TlvPresent](#)

## 8.255.1 Detailed Description

Structure for storing the NR5G Cell Access Status Info parameters

## Parameters

<i>nr5gCellStatus</i>	<ul style="list-style-type: none"> <li>Cell access status for NR5G calls. <ul style="list-style-type: none"> <li>Values: <ul style="list-style-type: none"> <li>NAS_CELL_ACCESS_NORMAL_ONLY (0x00) - Cell access is allowed for normal calls only</li> <li>NAS_CELL_ACCESS_EMERGENCY_ONLY (0x01) - Cell access is allowed for emergency calls only</li> <li>NAS_CELL_ACCESS_NO_CALLS (0x02) - Cell access is not allowed for any call type</li> <li>NAS_CELL_ACCESS_ALL_CALLS (0x03) - Cell access is allowed for all call types</li> <li>NAS_CELL_ACCESS_UNKNOWN (-1) - Cell access type is unknown</li> </ul> </li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present or not</li> </ul>

## 8.255.2 Field Documentation

8.255.2.1 uint32\_t nas\_NR5GCellStatusTlv::nr5gCellStatus

8.255.2.2 uint8\_t nas\_NR5GCellStatusTlv::TlvPresent

## 8.256 nas\_NR5GSerStatTlv Struct Reference

### Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [trueSrvStatus](#)
- uint8\_t [isPrefDataPath](#)
- uint8\_t [TlvPresent](#)

### 8.256.1 Detailed Description

This structure contains NR5G Service Status Info parameters.

#### Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> <li>• Service status of the system.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_STATUS_NO_SRV - No service</li> <li>– 0x01 - SYS_SRV_STATUS_LIMITED - Limited service</li> <li>– 0x02 - SYS_SRV_STATUS_SRV - Service</li> <li>– 0x03 - SYS_SRV_STATUS_LIMITED_REGIONAL - Limited regional service</li> <li>– 0x04 - SYS_SRV_STATUS_PWR_SAVE - Power save</li> </ul> </li> </ul>
<i>trueSrvStatus</i>	<ul style="list-style-type: none"> <li>• True service status of the system (not applicable to CDMA/HDR).</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - SYS_SRV_STATUS_NO_SRV - No service</li> <li>– 0x01 - SYS_SRV_STATUS_LIMITED - Limited service</li> <li>– 0x02 - SYS_SRV_STATUS_SRV - Service</li> <li>– 0x03 - SYS_SRV_STATUS_LIMITED_REGIONAL - Limited regional service</li> <li>– 0x04 - SYS_SRV_STATUS_PWR_SAVE - Power save</li> </ul> </li> </ul>
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> <li>• Whether the RAT is the preferred data path:</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Not preferred</li> <li>– 0x01 - Preferred</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present or not</li> </ul>

### 8.256.2 Field Documentation

8.256.2.1 uint8\_t nas\_NR5GSerStatTlv::isPrefDataPath

8.256.2.2 uint8\_t nas\_NR5GSerStatTlv::srvStatus

8.256.2.3 uint8\_t nas\_NR5GSerStatTlv::TlvPresent

8.256.2.4 uint8\_t nas\_NR5GSerStatTlv::trueSrvStatus

## 8.257 nas\_NR5GSrvStatusTlv Struct Reference

### Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [trueSrvStatus](#)
- uint8\_t [isPrefDataPath](#)
- uint8\_t [TLVPresent](#)

### 8.257.1 Detailed Description

Structure for storing the NR5G Service Status Info.

#### Parameters

<i>srvStatus</i>	<ul style="list-style-type: none"> <li>• Service status of the system.</li> <li>• Values <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> </ul> </li> </ul>
<i>trueSrvStatus</i>	<ul style="list-style-type: none"> <li>• True service status of the system (not applicable to CDMA/HDR).</li> <li>• Values <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> </ul> </li> </ul>
<i>isPrefDataPath</i>	<ul style="list-style-type: none"> <li>• Whether the RAT is the preferred data path</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Not preferred</li> <li>– 0x01 - Preferred</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.257.2 Field Documentation

8.257.2.1 uint8\_t nas\_NR5GSrvStatusTlv::isPrefDataPath



8.257.2.2 uint8\_t nas\_NR5GSrvStatusTlv::srvStatus

8.257.2.3 uint8\_t nas\_NR5GSrvStatusTlv::TLVPresent

8.257.2.4 uint8\_t nas\_NR5GSrvStatusTlv::trueSrvStatus

## 8.258 nas\_NR5GSysInfoTlv 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](#)
- uint8\_t [lacValid](#)
- uint16\_t [lac](#)
- uint8\_t [cellIdValid](#)
- uint32\_t [cellId](#)
- uint8\_t [regRejectInfoValid](#)
- uint8\_t [rejectSrvDomain](#)
- uint8\_t [rejectCause](#)
- uint8\_t [nwIdValid](#)
- uint8\_t [mcc](#) [3]
- uint8\_t [mnc](#) [3]
- uint8\_t [tacValid](#)
- uint16\_t [tac](#)
- uint8\_t [TLVPresent](#)

### 8.258.1 Detailed Description

Structure for storing the NR5G System Info.

#### Parameters

<i>srvDomainValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the service domain is valid.</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>– 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 - Camped</li> </ul> </li> </ul> </li> </ul>
<i>srvCapability-Valid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the service capability is valid.</li> </ul>

<i>srvCapability</i>	<ul style="list-style-type: none"> <li>• Current 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 - Camped</li> </ul> </li> </ul>
<i>roamStatusValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the roaming status is valid.</li> </ul>
<i>roamStatus</i>	<ul style="list-style-type: none"> <li>• Current roaming status.</li> <li>• Values <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> </ul> </li> <li>• Remainder of the values are per 3GPP2 C.R1001-F. 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.</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> </ul> </li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the location area code is valid.</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code (only applicable for 3GPP).</li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the cell ID is valid.</li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>• Cell ID.</li> </ul>

<i>regRejectInfo-Valid</i>	<ul style="list-style-type: none"> <li>Indicates whether the registration reject information is valid.</li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>Type of service domain in which the registration is rejected.</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 - Camped</li> </ul> </li> </ul>
<i>rejectCause</i>	<ul style="list-style-type: none"> <li>Reject cause values sent are specified in 3GPP TS 24.008 Sections 10.5.3.6 and 10.5.5.14, and 3GPP TS 24.301 Section 9.9.3.9.</li> </ul>
<i>nwldValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid.</li> </ul>
<i>mcc</i>	<ul style="list-style-type: none"> <li>MCC digits in ASCII characters.</li> <li>For CDMA, the MCC wildcard value is returned as {'3', 0xFF, 0xFF}.</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>MNC digits in ASCII characters.</li> <li>For this field <ul style="list-style-type: none"> <li>Unused byte is set to 0xFF</li> <li>In the 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 0x35 0x31 0xFF.</li> <li>For CDMA, the MNC wildcard value is returned as {'7', 0xFF, 0xFF}.</li> </ul> </li> </ul>
<i>tacValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the tracking area code is valid.</li> </ul>
<i>tac</i>	<ul style="list-style-type: none"> <li>Tracking area code (only applicable for NR5G).</li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>TLV Present</li> </ul>

## 8.258.2 Field Documentation

8.258.2.1 `uint32_t nas_NR5GSysInfoTlv::cellId`

8.258.2.2 `uint8_t nas_NR5GSysInfoTlv::cellIdValid`

8.258.2.3 `uint8_t nas_NR5GSysInfoTlv::isSysForbidden`

8.258.2.4 `uint8_t nas_NR5GSysInfoTlv::isSysForbiddenValid`

8.258.2.5 `uint16_t nas_NR5GSysInfoTlv::lac`

8.258.2.6 `uint8_t nas_NR5GSysInfoTlv::lacValid`

8.258.2.7 `uint8_t nas_NR5GSysInfoTlv::mcc[3]`

8.258.2.8 `uint8_t nas_NR5GSysInfoTlv::mnc[3]`

8.258.2.9 `uint8_t nas_NR5GSysInfoTlv::nwldValid`

8.258.2.10 `uint8_t nas_NR5GSysInfoTlv::regRejectInfoValid`

8.258.2.11 `uint8_t nas_NR5GSysInfoTlv::rejectCause`

8.258.2.12 `uint8_t nas_NR5GSysInfoTlv::rejectSrvDomain`

8.258.2.13 `uint8_t nas_NR5GSysInfoTlv::roamStatus`

8.258.2.14 `uint8_t nas_NR5GSysInfoTlv::roamStatusValid`

8.258.2.15 `uint8_t nas_NR5GSysInfoTlv::srvCapability`

8.258.2.16 `uint8_t nas_NR5GSysInfoTlv::srvCapabilityValid`

8.258.2.17 `uint8_t nas_NR5GSysInfoTlv::srvDomain`

8.258.2.18 `uint8_t nas_NR5GSysInfoTlv::srvDomainValid`

8.258.2.19 `uint16_t nas_NR5GSysInfoTlv::tac`

8.258.2.20 `uint8_t nas_NR5GSysInfoTlv::tacValid`

8.258.2.21 `uint8_t nas_NR5GSysInfoTlv::TLVPresent`

## 8.259 nas\_NR5GSystemInfoTlv Struct Reference

### Data Fields

- `uint8_t srvDomainValid`
- `uint8_t srvDomain`
- `uint8_t srvCapValid`
- `uint8_t srvcapability`
- `uint8_t roamStatusValid`
- `uint8_t roamStatus`
- `uint8_t sysForbiddenValid`
- `uint8_t sysForbidden`
- `uint8_t lacValid`
- `uint16_t lac`
- `uint8_t cellIdValid`
- `uint32_t cellId`
- `uint8_t regRejectInfoValid`
- `uint8_t rejectSrvDomain`
- `uint8_t rejCause`
- `uint8_t nwldValid`
- `int8_t MCC [3]`
- `int8_t MNC [3]`
- `uint8_t tacValid`

- uint16\_t [tac](#)
- uint8\_t [TlvPresent](#)

### 8.259.1 Detailed Description

This structure contains NR5G System Info parameters.

#### Parameters

<i>srvDomainValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the service domain is valid.</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>– Values:                 <ul style="list-style-type: none"> <li>* 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>* 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only</li> <li>* 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only</li> <li>* 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched</li> <li>* 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped</li> </ul> </li> </ul> </li> </ul>
<i>srvCapValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the service capability is valid.</li> </ul>
<i>srvCapability</i>	<ul style="list-style-type: none"> <li>• Current system service capability.             <ul style="list-style-type: none"> <li>– Values:                 <ul style="list-style-type: none"> <li>* 0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>* 0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only</li> <li>* 0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only</li> <li>* 0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched</li> <li>* 0x04 - SYS_SRV_DOMAIN_CAMPED - Camped</li> </ul> </li> </ul> </li> </ul>
<i>roamStatusValid</i>	<ul style="list-style-type: none"> <li>• Indicates whether the roaming status is valid.</li> </ul>

<i>roamStatus</i>	<ul style="list-style-type: none"> <li>Current roaming status. <ul style="list-style-type: none"> <li>Values: <ul style="list-style-type: none"> <li>* 0x00 - SYS_ROAM_STATUS_OFF - Off</li> <li>* 0x01 - SYS_ROAM_STATUS_ON - On</li> <li>* 0x02 - SYS_ROAM_STATUS_BLINK - Blinking</li> <li>* 0x03 - SYS_ROAM_STATUS_OUT_OF_NEIGHBORHOOD - Out of the neighborhood</li> <li>* 0x04 - SYS_ROAM_STATUS_OUT_OF_BLDG - Out of the building</li> <li>* 0x05 - SYS_ROAM_STATUS_PREF_SYS - Preferred system</li> <li>* 0x06 - SYS_ROAM_STATUS_AVAIL_SYS - Available system</li> <li>* 0x07 - SYS_ROAM_STATUS_ALLIANCE_PARTNER - Alliance partner</li> <li>* 0x08 - SYS_ROAM_STATUS_PREMIUM_PARTNER - Premium partner</li> <li>* 0x09 - SYS_ROAM_STATUS_FULL_SVC - Full service</li> <li>* 0x0A - SYS_ROAM_STATUS_PARTIAL_SVC -Partial service</li> <li>* 0x0B - SYS_ROAM_STATUS_BANNER_ON -Banner is on</li> <li>* 0x0C - SYS_ROAM_STATUS_BANNER_OFF -Banner is off Remainder of the values are per 3GPP2 C.R1001-F. Values from 0x02 onward are only applicable for 3GPP2.</li> </ul> </li> </ul> </li> </ul>
<i>sysForbidden-Valid</i>	<ul style="list-style-type: none"> <li>Indicates whether the forbidden system is valid.</li> </ul>
<i>sysForbidden</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> </ul> </li> </ul>
<i>lacValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the location area code is valid.</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>Location area code (only applicable for 3GPP).</li> </ul>
<i>cellIdValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the cell ID is valid.</li> </ul>
<i>cellId</i>	<ul style="list-style-type: none"> <li>Cell ID.</li> </ul>
<i>rejectSrvDomain</i>	<ul style="list-style-type: none"> <li>Type of service domain in which the registration is rejected.</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - SYS_SRV_DOMAIN_NO_SRV - No service</li> <li>0x01 - SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only</li> <li>0x02 - SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only</li> <li>0x03 - SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched</li> <li>0x04 - SYS_SRV_DOMAIN_CAMPED - Camped</li> </ul> </li> </ul>

<i>rejCause</i>	<ul style="list-style-type: none"> <li>Reject cause values sent are specified in 3GPP TS</li> <li>24.008 Sections 10.5.3.6 and 10.5.5.14, and 3GPP</li> <li>TS 24.301 Section 9.9.3.9.</li> </ul>
<i>nwldValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the network ID is valid.</li> </ul>
<i>MCC</i>	<ul style="list-style-type: none"> <li>MCC digits in ASCII characters. For CDMA, the MCC wildcard value is returned as {3, 0xFF, 0xFF}..</li> </ul>
<i>MNC</i>	<ul style="list-style-type: none"> <li>MNC digits in ASCII characters. For this field:</li> <li>Unused byte is set to 0xFF</li> <li>In the 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 0x35 0x31 0xFF. For CDMA, the MNC wildcard value is returned as {7, 0xFF, 0xFF}.</li> </ul>
<i>tacValid</i>	<ul style="list-style-type: none"> <li>Indicates whether the tracking area code is valid.</li> </ul>
<i>tac</i>	<ul style="list-style-type: none"> <li>Tracking area code (only applicable for NR5G).</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present or not</li> </ul>

## 8.259.2 Field Documentation

8.259.2.1 `uint32_t nas_NR5GSystemInfoTlv::cellId`

8.259.2.2 `uint8_t nas_NR5GSystemInfoTlv::cellIdValid`

8.259.2.3 `uint16_t nas_NR5GSystemInfoTlv::lac`

8.259.2.4 `uint8_t nas_NR5GSystemInfoTlv::lacValid`

8.259.2.5 `int8_t nas_NR5GSystemInfoTlv::MCC[3]`

8.259.2.6 `int8_t nas_NR5GSystemInfoTlv::MNC[3]`

8.259.2.7 `uint8_t nas_NR5GSystemInfoTlv::nwldValid`

8.259.2.8 `uint8_t nas_NR5GSystemInfoTlv::regRejectInfoValid`

8.259.2.9 `uint8_t nas_NR5GSystemInfoTlv::rejCause`

8.259.2.10 `uint8_t nas_NR5GSystemInfoTlv::rejectSrvDomain`

8.259.2.11 `uint8_t nas_NR5GSystemInfoTlv::roamStatus`

- 8.259.2.12 `uint8_t nas_NR5GSystemInfoTlv::roamStatusValid`
- 8.259.2.13 `uint8_t nas_NR5GSystemInfoTlv::srvcapability`
- 8.259.2.14 `uint8_t nas_NR5GSystemInfoTlv::srvCapValid`
- 8.259.2.15 `uint8_t nas_NR5GSystemInfoTlv::srvDomain`
- 8.259.2.16 `uint8_t nas_NR5GSystemInfoTlv::srvDomainValid`
- 8.259.2.17 `uint8_t nas_NR5GSystemInfoTlv::sysForbidden`
- 8.259.2.18 `uint8_t nas_NR5GSystemInfoTlv::sysForbiddenValid`
- 8.259.2.19 `uint16_t nas_NR5GSystemInfoTlv::tac`
- 8.259.2.20 `uint8_t nas_NR5GSystemInfoTlv::tacValid`
- 8.259.2.21 `uint8_t nas_NR5GSystemInfoTlv::TlvPresent`

## 8.260 nas\_NumScellsConfig Struct Reference

### Data Fields

- `uint16_t` [numScellsCfg](#)
- `uint8_t` [TlvPresent](#)

### 8.260.1 Detailed Description

This structure contains the parameters for Number of Scells Configured

#### Parameters

<i>numScellsCfg</i>	<ul style="list-style-type: none"> <li>• Number of Scells Configured.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.260.2 Field Documentation

- 8.260.2.1 `uint16_t nas_NumScellsConfig::numScellsCfg`
- 8.260.2.2 `uint8_t nas_NumScellsConfig::TlvPresent`

## 8.261 nas\_nwNameSrc3GppTlv Struct Reference

### Data Fields

- `uint32_t` [NwNameSrc3Gpp](#)
- `uint8_t` [TlvPresent](#)



### 8.261.1 Detailed Description

This structure contains the 3GPP Network Name Source parameters.

#### Parameters

<i>NwNameSrc3Gpp[OUT]</i>	<ul style="list-style-type: none"> <li>• 3GPP Network Name Source</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Unknown</li> <li>– 0x01 - Operator PLMN list and PLMN network name</li> <li>– 0x02 - Common PCN handset specification and operator name string</li> <li>– 0x03 - Network identity and time zone</li> <li>– 0x04 - GSMA SE13 table</li> <li>– 0x05 - Mobile country code and mobile network code</li> <li>– 0x06 - Service provider name</li> </ul> </li> </ul>
<i>TlvPresent[OUT]</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.261.2 Field Documentation

8.261.2.1 `uint32_t nas_nwNameSrc3GppTlv::NwNameSrc3Gpp`

8.261.2.2 `uint8_t nas_nwNameSrc3GppTlv::TlvPresent`

## 8.262 nas\_operatorNameString Struct Reference

#### Data Fields

- `uint8_t PLMNName` [255]

### 8.262.1 Detailed Description

This structure contains PLMN name.

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

### 8.262.2 Field Documentation

8.262.2.1 `uint8_t nas_operatorNameString::PLMNName[255]`

## 8.263 nas\_OperatorPLMNData Struct Reference

#### Data Fields

- `uint8_t mcc` [3]

- uint8\_t [mnc](#) [3]
- uint16\_t [lac1](#)
- uint16\_t [lac2](#)
- uint8\_t [PLMNRecID](#)

### 8.263.1 Detailed Description

This structure contains Operator PLMN.

#### 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>

### 8.263.2 Field Documentation

8.263.2.1 uint16\_t nas\_OperatorPLMNData::lac1

8.263.2.2 uint16\_t nas\_OperatorPLMNData::lac2

8.263.2.3 uint8\_t nas\_OperatorPLMNData::mcc[3]

8.263.2.4 uint8\_t nas\_OperatorPLMNData::mnc[3]

8.263.2.5 uint8\_t nas\_OperatorPLMNData::PLMNRecID

## 8.264 nas\_operatorPLMNList Struct Reference

#### Data Fields

- uint16\_t [numInstance](#)
- [nas\\_OperatorPLMNData](#) [PLMNData](#) [255]

### 8.264.1 Detailed Description

This structure contains Operator PLMN List.

## 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>

## 8.264.2 Field Documentation

8.264.2.1 uint16\_t nas\_operatorPLMNList::numInstance

8.264.2.2 nas\_OperatorPLMNData nas\_operatorPLMNList::PLMNData[255]

## 8.265 nas\_PhyCaAggDIBW Struct Reference

## Data Fields

- uint16\_t [aggDIBW](#)
- uint8\_t [TlvPresent](#)

## 8.265.1 Detailed Description

This structure contains the parameters for Aggregated DL Bandwidth.

## Parameters

<i>aggDIBW</i>	<ul style="list-style-type: none"> <li>Aggregated DL bandwidth.</li> <li>Units: MHz</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

## 8.265.2 Field Documentation

8.265.2.1 uint16\_t nas\_PhyCaAggDIBW::aggDIBW

8.265.2.2 uint8\_t nas\_PhyCaAggDIBW::TlvPresent

## 8.266 nas\_PhyCaAggPcellInfo Struct Reference

## Data Fields

- int [pci](#)
- int [freq](#)
- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) dl\_bw\_value
- int [iLTEbandValue](#)
- uint8\_t [TlvPresent](#)

### 8.266.1 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>

### 8.266.2 Field Documentation

8.266.2.1 [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) nas\_PhyCaAggPcellInfo::dl\_bw\_value

8.266.2.2 int nas\_PhyCaAggPcellInfo::freq

8.266.2.3 int nas\_PhyCaAggPcellInfo::iLTEbandValue

8.266.2.4 int nas\_PhyCaAggPcellInfo::pci

8.266.2.5 uint8\_t nas\_PhyCaAggPcellInfo::TlvPresent

## 8.267 nas\_PhyCaAggScellIDIBw Struct Reference

#### Data Fields

- [NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB\\_LITE](#) dl\_bw\_value
- uint8\_t [TlvPresent](#)

### 8.267.1 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>
--------------------	---

## 8.267.2 Field Documentation

8.267.2.1 NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE nas\_PhyCaAggScellIDBw::dl\_bw\_value

8.267.2.2 uint8\_t nas\_PhyCaAggScellIDBw::TlvPresent

## 8.268 nas\_PhyCaAggScellIndex Struct Reference

### Data Fields

- uint8\_t [scell\\_idx](#)
- uint8\_t [TlvPresent](#)

### 8.268.1 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>

## 8.268.2 Field Documentation

8.268.2.1 uint8\_t nas\_PhyCaAggScellIndex::scell\_idx

8.268.2.2 uint8\_t nas\_PhyCaAggScellIndex::TlvPresent

## 8.269 nas\_PhyCaAggScellIndType Struct Reference

### Data Fields

- int [pci](#)
- int [freq](#)
- NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE [scell\\_state](#)
- uint8\_t [TlvPresent](#)

### 8.269.1 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>

## 8.269.2 Field Documentation

8.269.2.1 int nas\_PhyCaAggScellIndType::freq

8.269.2.2 int nas\_PhyCaAggScellIndType::pci

8.269.2.3 NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE nas\_PhyCaAggScellIndType::scell\_state

8.269.2.4 uint8\_t nas\_PhyCaAggScellIndType::TlvPresent

## 8.270 nas\_PhyCaAggScellInfo Struct Reference

### Data Fields

- int [pci](#)
- int [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](#)

### 8.270.1 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>
-------------------	--

## 8.270.2 Field Documentation

8.270.2.1 **NAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE** nas\_PhyCaAggScellInfo::dl\_bw\_value

8.270.2.2 int nas\_PhyCaAggScellInfo::freq

8.270.2.3 uint16\_t nas\_PhyCaAggScellInfo::ltebandValue

8.270.2.4 int nas\_PhyCaAggScellInfo::pci

8.270.2.5 **NAS\_LTE\_CPHY\_SCELL\_STATE\_LITE** nas\_PhyCaAggScellInfo::scell\_state

8.270.2.6 uint8\_t nas\_PhyCaAggScellInfo::TlvPresent

## 8.271 nas\_PilotSetData Struct Reference

### Data Fields

- uint8\_t NumPilots
- nas\_PilotSetParams \* pPilotSetInfo

### 8.271.1 Detailed Description

This structure contains pilot sets data.

#### Parameters

<i>NumPilots</i>	<ul style="list-style-type: none"> <li>• Number of Pilot Sets</li> <li>• As input specifies number of sets of parameter pPilotSetInfo for which memory has been assigned</li> <li>• As output specifies the actual number of sets of parameter pPilotSetInfo returned by device</li> </ul>
<i>pPilotSetInfo</i>	<ul style="list-style-type: none"> <li>• Pilot Set Parameters</li> <li>• See <a href="#">nas_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.

## 8.271.2 Field Documentation

8.271.2.1 uint8\_t nas\_PilotSetData::NumPilots

8.271.2.2 **nas\_PilotSetParams\*** nas\_PilotSetData::pPilotSetInfo



## 8.272 nas\_PilotSetParams Struct Reference

### Data Fields

- uint32\_t [PilotType](#)
- uint16\_t [PilotPN](#)
- uint16\_t [PilotStrength](#)

### 8.272.1 Detailed Description

This structure contains pilot sets 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>

### 8.272.2 Field Documentation

8.272.2.1 uint16\_t nas\_PilotSetParams::PilotPN

8.272.2.2 uint16\_t nas\_PilotSetParams::PilotStrength

8.272.2.3 uint32\_t nas\_PilotSetParams::PilotType

## 8.273 nas\_PlmnID Struct Reference

### Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- uint8\_t [pcsDigit](#)
- uint8\_t [TlvPresent](#)

### 8.273.1 Detailed Description

This structure contains PLMN ID information

#### 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>pcsDigit</i>	<ul style="list-style-type: none"> <li>This field is used to interpret the length of the corresponding MNC reported in the TLVs (in this table) with an mnc or mobile_network_code field. Values: <ul style="list-style-type: none"> <li>– TRUE - MNC is a three-digit value; for example, a reported value of 90 corresponds to an MNC value of 090</li> <li>– FALSE - MNC is a two-digit value; for example, a reported value of 90 corresponds to an MNC value of 90</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present or not</li> </ul>

### 8.273.2 Field Documentation

8.273.2.1 uint16\_t nas\_PlmnID::mcc

8.273.2.2 uint16\_t nas\_PlmnID::mnc

8.273.2.3 uint8\_t nas\_PlmnID::pcsDigit

8.273.2.4 uint8\_t nas\_PlmnID::TlvPresent

## 8.274 nas\_PLMNNetworkName Struct Reference

### Data Fields

- uint8\_t numInstance
- nas\_PLMNNetworkNameData PLMNNetName [255]

### 8.274.1 Detailed Description

This structure contains NITZ Information list.

#### 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="#">nas_PLMNNetworkNameData</a> for details (Optional).</li> </ul>

### 8.274.2 Field Documentation

8.274.2.1 uint8\_t nas\_PLMNNetworkName::numInstance

8.274.2.2 nas\_PLMNNetworkNameData nas\_PLMNNetworkName::PLMNNetName[255]

## 8.275 nas\_PLMNNetworkNameData Struct Reference

## Data Fields

- uint8\_t [codingScheme](#)
- uint8\_t [countryInitials](#)
- uint8\_t [longNameSpareBits](#)
- uint8\_t [shortNameSpareBits](#)
- uint8\_t [longNameLen](#)
- uint8\_t [longName](#) [255]
- uint8\_t [shortNameLen](#)
- uint8\_t [shortName](#) [255]

## 8.275.1 Detailed Description

This structure contains NITZ Information.

## 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>

## 8.275.2 Field Documentation

8.275.2.1 uint8\_t nas\_PLMNNetworkNameData::codingScheme

8.275.2.2 uint8\_t nas\_PLMNNetworkNameData::countryInitials

8.275.2.3 uint8\_t nas\_PLMNNetworkNameData::longName[255]

8.275.2.4 uint8\_t nas\_PLMNNetworkNameData::longNameLen

8.275.2.5 uint8\_t nas\_PLMNNetworkNameData::longNameSpareBits

8.275.2.6 uint8\_t nas\_PLMNNetworkNameData::shortName[255]

8.275.2.7 uint8\_t nas\_PLMNNetworkNameData::shortNameLen

8.275.2.8 uint8\_t nas\_PLMNNetworkNameData::shortNameSpareBits

## 8.276 nas\_PRLPrefTlv Struct Reference

### Data Fields

- uint16\_t [PRLPref](#)
- uint8\_t [TlvPresent](#)

### 8.276.1 Detailed Description

Contain the PRL preference for system selection preferences.

#### Parameters

<i>PRLPref</i>	<ul style="list-style-type: none"> <li>• 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>TlvPresent</i>	<ul style="list-style-type: none"><li>• Tlv Present</li></ul>
-------------------	---

## 8.276.2 Field Documentation

8.276.2.1 uint16\_t nas\_PRLPrefTlv::PRLPref

8.276.2.2 uint8\_t nas\_PRLPrefTlv::TlvPresent

## 8.277 nas\_protocolSubtypeElement Struct Reference

### Data Fields

- uint16\_t [PhysicalLayer](#)
- uint16\_t [ControlMac](#)
- uint16\_t [AccessMac](#)
- uint16\_t [ForwardMac](#)
- uint16\_t [ReverseMac](#)
- uint16\_t [KeyExchange](#)
- uint16\_t [AuthProt](#)
- uint16\_t [EncryptProt](#)
- uint16\_t [SecProt](#)
- uint16\_t [IdleState](#)
- uint16\_t [MultDisc](#)
- uint16\_t [VirtStream](#)

### 8.277.1 Detailed Description

This structure contains protocol subtype element.

#### 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>

## 8.277.2 Field Documentation

8.277.2.1 uint16\_t nas\_protocolSubtypeElement::AccessMac

8.277.2.2 uint16\_t nas\_protocolSubtypeElement::AuthProt

8.277.2.3 uint16\_t nas\_protocolSubtypeElement::ControlMac

8.277.2.4 uint16\_t nas\_protocolSubtypeElement::EncryptProt

8.277.2.5 uint16\_t nas\_protocolSubtypeElement::ForwardMac

8.277.2.6 uint16\_t nas\_protocolSubtypeElement::IdleState

8.277.2.7 uint16\_t nas\_protocolSubtypeElement::KeyExchange

8.277.2.8 uint16\_t nas\_protocolSubtypeElement::MultDisc

8.277.2.9 uint16\_t nas\_protocolSubtypeElement::PhysicalLayer

8.277.2.10 uint16\_t nas\_protocolSubtypeElement::ReverseMac

8.277.2.11 uint16\_t nas\_protocolSubtypeElement::SecProt

8.277.2.12 uint16\_t nas\_protocolSubtypeElement::VirtStream

## 8.278 nas\_qaQmi3Gpp2TimeZone Struct Reference

### Data Fields

- uint8\_t [leapSeconds](#)
- uint8\_t [localTimeOffset](#)
- uint8\_t [daylightSavings](#)

### 8.278.1 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>

### 8.278.2 Field Documentation

8.278.2.1 uint8\_t nas\_qaQmi3Gpp2TimeZone::daylightSavings

8.278.2.2 uint8\_t nas\_qaQmi3Gpp2TimeZone::leapSeconds

8.278.2.3 uint8\_t nas\_qaQmi3Gpp2TimeZone::localTimeOffset

## 8.279 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]

### 8.279.1 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>

## 8.279.2 Field Documentation

8.279.2.1 `char nas_QmiNas3GppNetworkInfo::Description[255]`

8.279.2.2 `uint32_t nas_QmiNas3GppNetworkInfo::Forbidden`

8.279.2.3 `uint32_t nas_QmiNas3GppNetworkInfo::InUse`

8.279.2.4 `uint16_t nas_QmiNas3GppNetworkInfo::MCC`

8.279.2.5 `uint16_t nas_QmiNas3GppNetworkInfo::MNC`

8.279.2.6 `uint32_t nas_QmiNas3GppNetworkInfo::Preferred`

8.279.2.7 `uint32_t nas_QmiNas3GppNetworkInfo::Roaming`

## 8.280 nas\_QmiNas3GppNetworkRAT Struct Reference

### Data Fields

- `uint16_t` [MCC](#)
- `uint16_t` [MNC](#)
- `uint8_t` [RAT](#)



### 8.280.1 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>

### 8.280.2 Field Documentation

8.280.2.1 uint16\_t nas\_QmiNas3GppNetworkRAT::MCC

8.280.2.2 uint16\_t nas\_QmiNas3GppNetworkRAT::MNC

8.280.2.3 uint8\_t nas\_QmiNas3GppNetworkRAT::RAT

## 8.281 nas\_QmisNasPcsDigit Struct Reference

#### Data Fields

- uint16\_t [MCC](#)
- uint16\_t [MNC](#)
- uint8\_t [includes\\_pcs\\_digit](#)

### 8.281.1 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>

## 8.281.2 Field Documentation

8.281.2.1 `uint8_t nas_QmisNasPcsDigit::includes_pcs_digit`

8.281.2.2 `uint16_t nas_QmisNasPcsDigit::MCC`

8.281.2.3 `uint16_t nas_QmisNasPcsDigit::MNC`

## 8.282 `nas_QmisNasSlqsNasPCICellInfo` Struct Reference

### Data Fields

- `uint32_t freq`
- `uint16_t cellID`
- `uint32_t GlobalCellID`
- `uint8_t PlmnLen`
- `nas_QmisNasPcsDigit nasQmisNasPcsDigit` [255]

### 8.282.1 Detailed Description

Contain the PCI Cell Info.

#### Parameters

<i>freq</i>	<ul style="list-style-type: none"> <li>• Absolute cell's frequency. Range 0 to 65535.</li> </ul>
<i>cellID</i>	<ul style="list-style-type: none"> <li>• Cell ID.</li> </ul>
<i>GlobalCellID</i>	<ul style="list-style-type: none"> <li>• Global Cell ID.</li> </ul>
<i>PlmnLen</i>	<ul style="list-style-type: none"> <li>• Number of Plmn(<code>nasQmisNasPcsDigit</code>) information sets.</li> </ul>
<i>nasQmisNasPcsDigit</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_QmisNasPcsDigit</a> for more information.</li> </ul>

## 8.282.2 Field Documentation

8.282.2.1 `uint16_t nas_QmisNasSlqsNasPCICellInfo::cellID`

8.282.2.2 `uint32_t nas_QmisNasSlqsNasPCICellInfo::freq`

8.282.2.3 `uint32_t nas_QmisNasSlqsNasPCICellInfo::GlobalCellID`

8.282.2.4 `nas_QmisNasPcsDigit nas_QmisNasSlqsNasPCICellInfo::nasQmisNasPcsDigit`[255]

8.282.2.5 `uint8_t nas_QmisNasSlqsNasPCICellInfo::PlmnLen`

## 8.283 nas\_QmisNasSlqsNasPCIInfo Struct Reference

### Data Fields

- uint8\_t [PCICellInfoLen](#)
- [nas\\_QmisNasSlqsNasPCICellInfo](#) [nasQmisNasSlqsNasPCICellInfo](#) [255]
- uint16\_t [rsrp](#)
- uint16\_t [rsrpRx0](#)
- uint16\_t [rsrpRx1](#)
- uint16\_t [rsrq](#)
- uint16\_t [rsrqRx0](#)
- uint16\_t [rsrqRx1](#)

### 8.283.1 Detailed Description

Contain the NAS Perform network scan PCI information

#### Parameters

<i>PCICellInfoLen</i>	<ul style="list-style-type: none"> <li>• Number of PCI Cell Info(PCICellInfo) information sets.</li> </ul>
<i>nasQmisNasSlqsNasPCICellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_QmisNasSlqsNasPCICellInfo</a> for more information.</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>• Combined RSRP.</li> </ul>
<i>rsrpRx0</i>	<ul style="list-style-type: none"> <li>• Rx0 RSRP.</li> </ul>
<i>rsrpRx1</i>	<ul style="list-style-type: none"> <li>• Rx1 RSRP.</li> </ul>
<i>rsrq</i>	<ul style="list-style-type: none"> <li>• Combined RSRQ.</li> </ul>
<i>rsrqRx0</i>	<ul style="list-style-type: none"> <li>• Rx0 RSRQ.</li> </ul>
<i>rsrpRx1</i>	<ul style="list-style-type: none"> <li>• Rx1 RSRQ.</li> </ul>

### 8.283.2 Field Documentation

8.283.2.1 [nas\\_QmisNasSlqsNasPCICellInfo](#) [nas\\_QmisNasSlqsNasPCIInfo::nasQmisNasSlqsNasPCICellInfo](#)[255]

8.283.2.2 uint8\_t [nas\\_QmisNasSlqsNasPCIInfo::PCICellInfoLen](#)

8.283.2.3 uint16\_t [nas\\_QmisNasSlqsNasPCIInfo::rsrp](#)

8.283.2.4 uint16\_t [nas\\_QmisNasSlqsNasPCIInfo::rsrpRx0](#)

8.283.2.5 uint16\_t nas\_QmisNasSlqsNasPCInfo::rsrpRx1

8.283.2.6 uint16\_t nas\_QmisNasSlqsNasPCInfo::rsrq

8.283.2.7 uint16\_t nas\_QmisNasSlqsNasPCInfo::rsrqRx0

8.283.2.8 uint16\_t nas\_QmisNasSlqsNasPCInfo::rsrqRx1

## 8.284 nas\_RankIndicatorTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [count0](#)
- uint32\_t [count1](#)

### 8.284.1 Detailed Description

Elements for Rank indicator TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if TLV is present, otherwise 0</li> </ul>
<i>count0</i>	<ul style="list-style-type: none"> <li>• count of Rank1</li> </ul>
<i>count1</i>	<ul style="list-style-type: none"> <li>• count of Rank2</li> </ul>

### 8.284.2 Field Documentation

8.284.2.1 uint32\_t nas\_RankIndicatorTlv::count0

8.284.2.2 uint32\_t nas\_RankIndicatorTlv::count1

8.284.2.3 uint8\_t nas\_RankIndicatorTlv::TlvPresent

## 8.285 nas\_RatDisabledMaskTlv Struct Reference

### Data Fields

- uint16\_t [ratDisabledMask](#)
- uint8\_t [TlvPresent](#)

### 8.285.1 Detailed Description

Contain the RAT disabled mask

## Parameters

<i>ratDisabledMask</i>	<ul style="list-style-type: none"> <li>• Bitmask representing the radio technologies that are disabled.</li> <li>• Bit values: <ul style="list-style-type: none"> <li>– Bit 0 (0x01) - cdma2000 @ 1X</li> <li>– Bit 1 (0x02) - cdma2000 @ HRPD(1xEV-DO)</li> <li>– Bit 2 (0x04) - GSM</li> <li>– Bit 3 (0x08) - UMTS</li> <li>– Bit 4 (0x10) - LTE</li> <li>– Bit 5 (0x20) - TD-SCDMA</li> <li>– Bit 6 (0x40) - NR5G All unlisted bits are reserved for future use and the service point ignores them if used.</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.285.2 Field Documentation

8.285.2.1 uint16\_t nas\_RatDisabledMaskTlv::ratDisabledMask

8.285.2.2 uint8\_t nas\_RatDisabledMaskTlv::TlvPresent

## 8.286 nas\_RejectReasonTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [serviceDomain](#)
- uint32\_t [rejectCause](#)

## 8.286.1 Detailed Description

This structure contains paramaters reject reason.

## 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

## 8.286.2 Field Documentation

8.286.2.1 uint32\_t nas\_RejectReasonTlv::rejectCause

8.286.2.2 uint32\_t nas\_RejectReasonTlv::serviceDomain

8.286.2.3 uint8\_t nas\_RejectReasonTlv::TlvPresent

## 8.287 nas\_RFBandInfoElements Struct Reference

## Data Fields

- uint8\_t [radioInterface](#)
- uint16\_t [activeBandClass](#)
- uint16\_t [activeChannel](#)

### 8.287.1 Detailed Description

This structure contains the RFBandInfo response parameters.

#### Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface technology <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Radio Interface</li> </ul> </li> </ul>
<i>activeBandClass</i>	<ul style="list-style-type: none"> <li>• Active Band Class <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Band Classes</li> </ul> </li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>• Active channel (0 if channel is not relevant to the reported technology)</li> </ul>

### 8.287.2 Field Documentation

8.287.2.1 uint16\_t nas\_RFBandInfoElements::activeBandClass

8.287.2.2 uint16\_t nas\_RFBandInfoElements::activeChannel

8.287.2.3 uint8\_t nas\_RFBandInfoElements::radioInterface

## 8.288 nas\_RfBandInfoExtFormat Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t \* [pInstancesSize](#)
- struct [nas\\_RfBandInfoExtFormatElements](#) \* [pRfBandInfoExtFormatParam](#)

### 8.288.1 Detailed Description

This structure contains the [nas\\_RfBandInfoExtFormat](#) response parameters.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<i>pInstancesSize[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Number of sets of the <a href="#">RfBandInfoElements</a></li> <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>pRfBandInfoExtFormatParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RfBandInfoExtFormatElements</a> for more information</li> </ul>

## 8.288.2 Field Documentation

8.288.2.1 `uint8_t* nas_RfBandInfoExtFormat::pInstancesSize`8.288.2.2 `struct nas_RfBandInfoExtFormatElements* nas_RfBandInfoExtFormat::pRfBandInfoExtFormatParam`8.288.2.3 `uint8_t nas_RfBandInfoExtFormat::TlvPresent`

## 8.289 nas\_RfBandInfoExtFormatElements Struct Reference

## Data Fields

- `uint8_t` [radioInterface](#)
- `uint16_t` [activeBand](#)
- `uint32_t` [activeChannel](#)

## 8.289.1 Detailed Description

This structure contains the RF Band Information, Extended Format.

## 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>activeBand</i>	<ul style="list-style-type: none"> <li>• Active Band Class <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Band Classes</li> </ul> </li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>• Active channel (0 if channel is not relevant to the reported technology)</li> </ul>

## 8.289.2 Field Documentation

8.289.2.1 `uint16_t nas_RfBandInfoExtFormatElements::activeBand`8.289.2.2 `uint32_t nas_RfBandInfoExtFormatElements::activeChannel`

8.289.2.3 uint8\_t nas\_RfBandInfoExtFormatElements::radioInterface

## 8.290 nas\_RFBandInfoExtTlv Struct Reference

### Data Fields

- uint8\_t [radioIrf](#)
- uint16\_t [activeBand](#)
- uint32\_t [activeChannel](#)
- uint8\_t [TlvPresent](#)

### 8.290.1 Detailed Description

Contain fields in struct [nas\\_RFBandInfoExtTlv](#)

#### Parameters

<i>radioIrf</i>	<ul style="list-style-type: none"> <li>• Radio interface currently in use.</li> <li>• Values <ul style="list-style-type: none"> <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> <li>– 0x09 - TD-SCDMA</li> <li>– 0x0C - NR5G</li> </ul> </li> </ul>
<i>activeBand</i>	<ul style="list-style-type: none"> <li>• Active band class.</li> <li>• Values <ul style="list-style-type: none"> <li>– 00 to 39 - CDMA band classes</li> <li>– 40 to 79 - GSM band classes</li> <li>– 80 to 91 - WCDMA band classes</li> <li>– 120 to 168 - LTE band classes</li> <li>– 200 to 205 - TD-SCDMA band classes</li> <li>– 250 to 251 - NR-5G band classes</li> </ul> </li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>• Active channel.</li> <li>• If the channel is not relevant to the technology, a value of 0 is returned.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

#### Note

None

### 8.290.2 Field Documentation



8.290.2.1 uint16\_t nas\_RFBandInfoExtTlv::activeBand

8.290.2.2 uint32\_t nas\_RFBandInfoExtTlv::activeChannel

8.290.2.3 uint8\_t nas\_RFBandInfoExtTlv::radiolf

8.290.2.4 uint8\_t nas\_RFBandInfoExtTlv::TlvPresent

## 8.291 nas\_RfBandInfoList Struct Reference

### Data Fields

- uint8\_t \* [pInstanceSize](#)
- struct [nas\\_RFBandInfoElements](#) \* [pRFBandInfoParam](#)

### 8.291.1 Detailed Description

This structure contains the RfBandInfoList response parameters.

#### Parameters

<i>pInstanceSize</i> [I- N/OUT]	<ul style="list-style-type: none"> <li>• Number of sets of the <a href="#">RFBandInfoElements</a></li> <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>pRFBandInfo- Param</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RFBandInfoElements</a> for more information</li> </ul>

### 8.291.2 Field Documentation

8.291.2.1 uint8\_t\* nas\_RfBandInfoList::pInstanceSize

8.291.2.2 struct [nas\\_RFBandInfoElements](#)\* nas\_RfBandInfoList::pRFBandInfoParam

## 8.292 nas\_RFBandInfoTlv Struct Reference

### Data Fields

- uint8\_t [radiolf](#)
- uint16\_t [activeBand](#)
- uint16\_t [activeChannel](#)
- uint8\_t [TlvPresent](#)

### 8.292.1 Detailed Description

Contain fields in struct [nas\\_RFBandInfoTlv](#)

## Parameters

<i>radioIrf</i>	<ul style="list-style-type: none"> <li>• Radio interface currently in use.</li> <li>• Values <ul style="list-style-type: none"> <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> <li>– 0x09 - TD-SCDMA</li> <li>– 0x0C - NR5G</li> </ul> </li> </ul>
<i>activeBand</i>	<ul style="list-style-type: none"> <li>• Active band class.</li> <li>• Values <ul style="list-style-type: none"> <li>– 00 to 39 - CDMA band classes</li> <li>– 40 to 79 - GSM band classes</li> <li>– 80 to 91 - WCDMA band classes</li> <li>– 120 to 168 - LTE band classes</li> <li>– 200 to 205 - TD-SCDMA band classes</li> <li>– 250 to 251 - NR-5G band classes</li> </ul> </li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>• Active channel.</li> <li>• If the channel is not relevant to the technology, a value of 0 is returned.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## Note

None

## 8.292.2 Field Documentation

8.292.2.1 uint16\_t nas\_RFBandInfoTlv::activeBand

8.292.2.2 uint16\_t nas\_RFBandInfoTlv::activeChannel

8.292.2.3 uint8\_t nas\_RFBandInfoTlv::radioIrf

8.292.2.4 uint8\_t nas\_RFBandInfoTlv::TlvPresent

## 8.293 nas\_RfBandwidthInfo Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t \* [pInstancesSize](#)
- struct [nas\\_RfBandwidthInfoElements](#) \* [pRfBandwidthInfoParam](#)

### 8.293.1 Detailed Description

This structure contains the [nas\\_RfBandwidthInfo](#) response parameters.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<i>pInstancesSize[IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Number of sets of the <a href="#">RFBandInfoElements</a></li> <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>pRfBandwidthInfoParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RfBandwidthInfoElements</a> for more information</li> </ul>

### 8.293.2 Field Documentation

8.293.2.1 `uint8_t* nas_RfBandwidthInfo::pInstancesSize`

8.293.2.2 `struct nas_RfBandwidthInfoElements* nas_RfBandwidthInfo::pRfBandwidthInfoParam`

8.293.2.3 `uint8_t nas_RfBandwidthInfo::TlvPresent`

## 8.294 nas\_RfBandwidthInfoElements Struct Reference

### Data Fields

- `uint8_t` [radioInterface](#)
- `uint32_t` [bandwidth](#)

### 8.294.1 Detailed Description

This structure contains the RF Bandwidth Information List 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>bandwidth</i>	<ul style="list-style-type: none"> <li>Bandwidth <ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>NAS_LTE_BW_NRB_6 (0) - LTE 1.4 MHz bandwidth</li> <li>NAS_LTE_BW_NRB_15 (1) - LTE 3 MHz bandwidth</li> <li>NAS_LTE_BW_NRB_25 (2) - LTE 5 MHz bandwidth</li> <li>NAS_LTE_BW_NRB_50 (3) - LTE 10 MHz bandwidth</li> <li>NAS_LTE_BW_NRB_75 (4) - LTE 15 MHz bandwidth</li> <li>NAS_LTE_BW_NRB_100 (5) - LTE 20 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_6 (6) - NR5G 1.4 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_15 (7) - NR5G 3 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_25 (8) - NR5G 5 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_50 (9) - NR5G 10 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_75 (10) - NR5G 15 MHz bandwidth</li> <li>NAS_NR5G_BW_NRB_100 (11) - NR5G 20 MHz bandwidth</li> </ul> </li> </ul> </li> </ul>

## 8.294.2 Field Documentation

8.294.2.1 uint32\_t nas\_RfBandwidthInfoElements::bandwidth

8.294.2.2 uint8\_t nas\_RfBandwidthInfoElements::radioInterface

## 8.295 nas\_RFBandwidthInfoTlv Struct Reference

## Data Fields

- uint8\_t [radioIf](#)
- uint32\_t [bandwidth](#)
- uint8\_t [TlvPresent](#)

## 8.295.1 Detailed Description

Contain fields in struct [nas\\_RFBandwidthInfoTlv](#)

## Parameters

<i>radioIrf</i>	<ul style="list-style-type: none"> <li>• Radio interface currently in use.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - 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>– 0x06 - WLAN</li> <li>– 0x07 - GPS</li> <li>– 0x08 - LTE</li> <li>– 0x09 - TD-SCDMA</li> <li>– 0x0a - LTE-M1</li> <li>– 0x0b - LTE-NB1</li> <li>– -1 - No change</li> </ul> </li> </ul>
<i>bandwidth</i>	<ul style="list-style-type: none"> <li>• Bandwidth.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - LTE 1.4 MHz bandwidth</li> <li>– 1 - LTE 3 MHz bandwidth</li> <li>– 2 - LTE 5 MHz bandwidth</li> <li>– 3 - LTE 10 MHz bandwidth</li> <li>– 4 - LTE 15 MHz bandwidth</li> <li>– 5 - LTE 20 MHz bandwidth</li> <li>– 6 - NR5G 1.4 MHz bandwidth</li> <li>– 7 - NR5G 3 MHz bandwidth</li> <li>– 8 - NR5G 5 MHz bandwidth</li> <li>– 9 - NR5G 10 MHz bandwidth</li> <li>– 10 - NR5G 15 MHz bandwidth</li> <li>– 11 - NR5G 20 MHz bandwidth</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## Note

None

## 8.295.2 Field Documentation

8.295.2.1 uint32\_t nas\_RFBandwidthInfoTlv::bandwidth

8.295.2.2 uint8\_t nas\_RFBandwidthInfoTlv::radioIrf

8.295.2.3 uint8\_t nas\_RFBandwidthInfoTlv::TlvPresent

## 8.296 nas\_RfDedicatedBandInfo Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t \* [pInstancesSize](#)
- struct [nas\\_RfDedicatedBandInfoElements](#) \* [pRfDedicatedBandInfoParam](#)

### 8.296.1 Detailed Description

This structure contains the [nas\\_RfDedicatedBandInfo](#) response parameters.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>
<i>pInstancesSize[-IN/OUT]</i>	<ul style="list-style-type: none"> <li>• Number of sets of the <a href="#">RFBandInfoElements</a></li> <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>pRfDedicated-BandInfoParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RfDedicatedBandInfoElements</a> for more information</li> </ul>

### 8.296.2 Field Documentation

8.296.2.1 uint8\_t\* [nas\\_RfDedicatedBandInfo::pInstancesSize](#)

8.296.2.2 struct [nas\\_RfDedicatedBandInfoElements](#)\* [nas\\_RfDedicatedBandInfo::pRfDedicatedBandInfoParam](#)

8.296.2.3 uint8\_t [nas\\_RfDedicatedBandInfo::TlvPresent](#)

## 8.297 nas\_RfDedicatedBandInfoElements Struct Reference

## Data Fields

- uint8\_t [radioInterface](#)
- uint16\_t [dedicatedBand](#)

### 8.297.1 Detailed Description

This structure contains the RF Dedicated Band Info.

#### 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>dedicatedBand</i>	<ul style="list-style-type: none"> <li>• Dedicated Band Class             <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Band Classes</li> </ul> </li> </ul>

## 8.297.2 Field Documentation

8.297.2.1 uint16\_t nas\_RfDedicatedBandInfoElements::dedicatedBand

8.297.2.2 uint8\_t nas\_RfDedicatedBandInfoElements::radioInterface

## 8.298 nas\_RFDedicatedBandInfoTlv Struct Reference

### Data Fields

- uint8\_t [radioInterface](#)
- uint16\_t [dedicatedBand](#)
- uint8\_t [TlvPresent](#)

### 8.298.1 Detailed Description

Contain fields in struct [nas\\_RFDedicatedBandInfoTlv](#)

#### Parameters

<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 - 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> <li>– 0x09 - TD-SCDMA</li> </ul> </li> </ul>
<i>dedicatedBand</i>	<ul style="list-style-type: none"> <li>• Dedicated band class.</li> <li>• Values <ul style="list-style-type: none"> <li>– 00 to 39 - CDMA band classes</li> <li>– 40 to 79 - GSM band classes</li> <li>– 80 to 91 - WCDMA band classes</li> <li>– 120 to 168 - LTE band classes</li> <li>– 200 to 205 - TD-SCDMA band classes</li> <li>– 0xFFFF - invalid - indicates that the UE moved out from the dedicated band</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

#### Note

None

## 8.298.2 Field Documentation

8.298.2.1 uint16\_t nas\_RFDedicatedBandInfoTlv::dedicatedBand

8.298.2.2 `uint8_t nas_RFDedicatedBandInfoTlv::radioIrf`

8.298.2.3 `uint8_t nas_RFDedicatedBandInfoTlv::TlvPresent`

## 8.299 nas\_RFInfoTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t radioInterfaceSize`
- `uint32_t radioInterface` [255]
- `uint32_t activeBandClass` [255]
- `uint32_t activeChannel` [255]

### 8.299.1 Detailed Description

This structure contains parameters RF Band Information List.

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>radioInterfaceSize</i>	Number of sets of the following elements <ul style="list-style-type: none"> <li>• <code>radioInterface</code></li> <li>• <code>activeBandClass</code></li> <li>• <code>activeChannel</code></li> </ul>
<i>radioInterface</i>	radio interface technology of the signal being measured <ul style="list-style-type: none"> <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> <li>• 0x09 - TD-SCDMA</li> </ul>
<i>activeBandClass</i>	active band class <ul style="list-style-type: none"> <li>• 00 to 39 - CDMA band classes</li> <li>• 40 to 79 - GSM band classes</li> <li>• 80 to 91 - WCDMA band classes</li> <li>• 120 to 160 - LTE band classes</li> <li>• 200 to 205 - TD-SCDMA band classes</li> </ul>
<i>activeChannel</i>	active channel <ul style="list-style-type: none"> <li>• Active channel. If the channel is not relevant to the technology, a value of 0 is returned.</li> </ul>

### 8.299.2 Field Documentation

8.299.2.1 `uint32_t nas_RFInfoTlv::activeBandClass[255]`

8.299.2.2 `uint32_t nas_RFInfoTlv::activeChannel[255]`



8.299.2.3 uint32\_t nas\_RFInfoTlv::radioInterface[255]

8.299.2.4 uint8\_t nas\_RFInfoTlv::radioInterfaceSize

8.299.2.5 uint8\_t nas\_RFInfoTlv::TlvPresent

## 8.300 nas\_roamIndList Struct Reference

### Data Fields

- uint8\_t [numInstances](#)
- uint8\_t [radioInterface](#) [32]
- uint8\_t [roamIndicator](#) [32]

### 8.300.1 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_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>
<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>

### 8.300.2 Field Documentation

8.300.2.1 uint8\_t nas\_roamIndList::numInstances

8.300.2.2 uint8\_t nas\_roamIndList::radioInterface[32]

8.300.2.3 uint8\_t nas\_roamIndList::roamIndicator[32]

## 8.301 nas\_RoamPrefTlv Struct Reference

### Data Fields

- uint16\_t [RoamPref](#)
- uint8\_t [TlvPresent](#)

### 8.301.1 Detailed Description

Contain the roaming preference for system selection preferences.

#### Parameters

<i>RoamPref</i>	<ul style="list-style-type: none"> <li>• 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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.301.2 Field Documentation

8.301.2.1 uint16\_t nas\_RoamPrefTlv::RoamPref

8.301.2.2 uint8\_t nas\_RoamPrefTlv::TlvPresent

## 8.302 nas\_RSRPThresh Struct Reference

### Data Fields

- uint8\_t [RSRPThresListLen](#)
- int16\_t \* [pRSRPThresList](#)

### 8.302.1 Detailed Description

This structure contains RSRP threshold list.

## Parameters

<i>RSRPThresListLen</i>	<ul style="list-style-type: none"> <li>Length of the 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>

## 8.302.2 Field Documentation

8.302.2.1 int16\_t\* nas\_RSRPThresh::pRSRPThresList

8.302.2.2 uint8\_t nas\_RSRPThresh::RSRPThresListLen

## 8.303 nas\_rsrqInformation Struct Reference

## Data Fields

- int8\_t [rsrq](#)
- uint8\_t [radiolf](#)

## 8.303.1 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>

## 8.303.2 Field Documentation

8.303.2.1 uint8\_t nas\_rsrqInformation::radiolf

8.303.2.2 int8\_t nas\_rsrqInformation::rsrq

## 8.304 nas\_RSRQThresh Struct Reference

## Data Fields

- uint8\_t [RSRQThresListLen](#)
- int8\_t \* [pRSRQThresList](#)

### 8.304.1 Detailed Description

This structure contains RSRQ threshold list.

#### Parameters

<i>RSRQThresListLen</i>	<ul style="list-style-type: none"> <li>• Length of the 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>

### 8.304.2 Field Documentation

8.304.2.1 int8\_t\* nas\_RSRQThresh::pRSRQThresList

8.304.2.2 uint8\_t nas\_RSRQThresh::RSRQThresListLen

## 8.305 nas\_RSSIThresh Struct Reference

## Data Fields

- uint8\_t [RSSIThresListLen](#)
- int8\_t \* [pRSSIThresList](#)

### 8.305.1 Detailed Description

This sturcture contain RSSI threshold list.

## 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>

## 8.305.2 Field Documentation

8.305.2.1 int8\_t\* nas\_RSSIThresh::pRSSIThresList

8.305.2.2 uint8\_t nas\_RSSIThresh::RSSIThresListLen

## 8.306 nas\_rxInfo Struct Reference

## Data Fields

- uint8\_t [isRadioTuned](#)
- int32\_t [rxPower](#)
- int32\_t [ecio](#)
- int32\_t [rscp](#)
- int32\_t [rsrp](#)
- uint32\_t [phase](#)

## 8.306.1 Detailed Description

This structure contains rx radio 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 dB; 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>

## 8.306.2 Field Documentation

8.306.2.1 `int32_t nas_rxInfo::ecio`

8.306.2.2 `uint8_t nas_rxInfo::isRadioTuned`

8.306.2.3 `uint32_t nas_rxInfo::phase`

8.306.2.4 `int32_t nas_rxInfo::rscp`

8.306.2.5 `int32_t nas_rxInfo::rsrp`

8.306.2.6 `int32_t nas_rxInfo::rxPower`

## 8.307 nas\_RxSigInfo Struct Reference

### Data Fields

- `uint8_t rxChainIndex`
- `uint8_t isRadioTuned`
- `int32_t rxPower`
- `int32_t rsrp`

### 8.307.1 Detailed Description

This structure contains the parameters for Rx Signal Info.

#### Parameters

<i>rxChainIndex</i>	<ul style="list-style-type: none"> <li>• Rx antenna path</li> <li>• Valid Values <ul style="list-style-type: none"> <li>– 0 - Primary Rx</li> <li>– 1 - Diversity Rx</li> </ul> </li> </ul>
---------------------	---

<i>isRadioTuned</i>	<ul style="list-style-type: none"> <li>Rx path is tuned to a channel or Not</li> <li>Values <ul style="list-style-type: none"> <li>0x00 - Not tuned</li> <li>0x01 - Tuned</li> </ul> </li> </ul>
---------------------	--

**Note**

If the radio is tuned, the instantaneous values are set for the fields below. If the radio is not tuned, the values set below may be invalid.

**Parameters**

<i>rxPower</i>	<ul style="list-style-type: none"> <li>Rx power value in 1/10 dBm resolution</li> </ul>
<i>rsrp</i>	<ul style="list-style-type: none"> <li>Current reference signal received power in 1/10 dBm resolution</li> </ul>

**8.307.2 Field Documentation**

8.307.2.1 `uint8_t nas_RxSigInfo::isRadioTuned`

8.307.2.2 `int32_t nas_RxSigInfo::rsrp`

8.307.2.3 `uint8_t nas_RxSigInfo::rxChainIndex`

8.307.2.4 `int32_t nas_RxSigInfo::rxPower`

**8.308 nas\_rxSignalStrengthListElement Struct Reference****Data Fields**

- `int16_t rxSignalStrength`
- `uint8_t radiolf`

**8.308.1 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.

**8.308.2 Field Documentation**

8.308.2.1 `uint8_t nas_rxSignalStrengthListElement::radioIf`

8.308.2.2 `int16_t nas_rxSignalStrengthListElement::rxSignalStrength`

**8.309 nas\_SccRxInfo Struct Reference****Data Fields**

- `int32_t rsrq`
- `int16_t snr`
- `uint8_t numInstances`
- `nas_RxSigInfo sigInfo` [255]
- `uint8_t TlvPresent`

**8.309.1 Detailed Description**

This structure contains information about the SccRxInfo parameters.

**Parameters**

<i>rsrq</i>	<ul style="list-style-type: none"> <li>Current reference signal</li> <li>Receive quality in 1/10 dB resolution</li> </ul>
<i>snr</i>	<ul style="list-style-type: none"> <li>Reference signal signal-to-noise ratio in dB.</li> <li>Range -10 to 30</li> </ul>
<i>numInstances</i>	<ul style="list-style-type: none"> <li>Number of sets of the following             <ul style="list-style-type: none"> <li>rxChainIndex</li> <li>isRadioTuned</li> <li>rxPower</li> <li>rsrp</li> </ul> </li> </ul>



<i>sigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RxSigInfo</a> for more information</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.309.2 Field Documentation

8.309.2.1 `uint8_t nas_SccRxInfo::numInstances`

8.309.2.2 `int32_t nas_SccRxInfo::rsrq`

8.309.2.3 `nas_RxSigInfo nas_SccRxInfo::sigInfo[255]`

8.309.2.4 `int16_t nas_SccRxInfo::snr`

8.309.2.5 `uint8_t nas_SccRxInfo::TlvPresent`

## 8.310 nas\_serviceProviderName Struct Reference

### Data Fields

- `uint8_t displayCondition`
- `uint8_t spnLength`
- `uint8_t spn [255]`

### 8.310.1 Detailed Description

This structure contains service provider name.

#### 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>

## 8.310.2 Field Documentation

8.310.2.1 `uint8_t nas_serviceProviderName::displayCondition`

8.310.2.2 `uint8_t nas_serviceProviderName::spn[255]`

8.310.2.3 `uint8_t nas_serviceProviderName::spnLength`

## 8.311 nas\_servSystem Struct Reference

### Data Fields

- uint8\_t [regState](#)
- uint8\_t [csAttachState](#)
- uint8\_t [psAttachState](#)
- uint8\_t [selNetwork](#)
- uint8\_t [numRadioInterfaces](#)
- uint8\_t [radioInterface](#) [32]

### 8.311.1 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_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>
-----------------------	--

### 8.311.2 Field Documentation

8.311.2.1 `uint8_t nas_servSystem::csAttachState`

8.311.2.2 `uint8_t nas_servSystem::numRadioInterfaces`

8.311.2.3 `uint8_t nas_servSystem::psAttachState`

8.311.2.4 `uint8_t nas_servSystem::radioInterface[32]`

8.311.2.5 `uint8_t nas_servSystem::regState`

8.311.2.6 `uint8_t nas_servSystem::selNetwork`

## 8.312 nas\_sidNid Struct Reference

### Data Fields

- `uint16_t` [nid](#)
- `uint16_t` [sid](#)

### 8.312.1 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>

### 8.312.2 Field Documentation

8.312.2.1 `uint16_t nas_sidNid::nid`

8.312.2.2 `uint16_t nas_sidNid::sid`

## 8.313 nas\_SignalStrengthTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- int8\_t [signalStrength](#)
- uint32\_t [radioInterface](#)

### 8.313.1 Detailed Description

This structure contains paramaters Signal Strength.

#### 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

### 8.313.2 Field Documentation

8.313.2.1 uint32\_t nas\_SignalStrengthTlv::radioInterface

8.313.2.2 int8\_t nas\_SignalStrengthTlv::signalStrength

8.313.2.3 uint8\_t nas\_SignalStrengthTlv::TlvPresent

## 8.314 nas\_SimRejInfoTlv Struct Reference

### Data Fields

- uint32\_t [SimRejInfo](#)
- uint8\_t [TLVPresent](#)

### 8.314.1 Detailed Description

Structure for storing the SIM Reject Information.

#### Parameters

<i>SimRejInfo</i>	<ul style="list-style-type: none"> <li>• Current reject state information of the SIM.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - SIM is not available</li> <li>– 1 - SIM is available</li> <li>– 2 - SIM has been marked by the network as invalid for circuit-switched services</li> <li>– 3 - SIM has been marked by the network as invalid for packet-switched services</li> <li>– 4 - SIM has been marked by the network as invalid for circuit-switched and packet-switched services</li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>• TLV Present</li> </ul>

### 8.314.2 Field Documentation

8.314.2.1 uint32\_t nas\_SimRejInfoTlv::SimRejInfo

8.314.2.2 uint8\_t nas\_SimRejInfoTlv::TLVPresent

## 8.315 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

### 8.315.1 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>ltesrpdelta</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>

## 8.315.2 Field Documentation

8.315.2.1 uint8\_t nas\_SLQSSignalStrengthsIndReq::ecioDelta

8.315.2.2 int16\_t nas\_SLQSSignalStrengthsIndReq::ecioThresholdList[10]

8.315.2.3 uint8\_t nas\_SLQSSignalStrengthsIndReq::ecioThresholdListLen

8.315.2.4 uint8\_t nas\_SLQSSignalStrengthsIndReq::ioDelta

8.315.2.5 uint8\_t nas\_SLQSSignalStrengthsIndReq::lteRsrpDelta

8.315.2.6 uint16\_t nas\_SLQSSignalStrengthsIndReq::lteSnrDelta

8.315.2.7 uint8\_t nas\_SLQSSignalStrengthsIndReq::rsrqDelta

8.315.2.8 uint8\_t nas\_SLQSSignalStrengthsIndReq::rxSignalStrengthDelta

8.315.2.9 uint8\_t nas\_SLQSSignalStrengthsIndReq::sinrDelta

8.315.2.10 uint8\_t nas\_SLQSSignalStrengthsIndReq::sinrThresholdList[5]

8.315.2.11 uint8\_t nas\_SLQSSignalStrengthsIndReq::sinrThresholdListLen

## 8.316 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](#)

### 8.316.1 Detailed Description

This structure contains parameters signal strengths information.

#### Parameters

<i>rxSignalStrengthInfo</i>	<ul style="list-style-type: none"> <li>• signal strength info list</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ecioInfo</i>	<ul style="list-style-type: none"> <li>• ecio info list</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>io</i>	<ul style="list-style-type: none"> <li>• received IO in dBm; IO is only applicable for 1xEV-DO</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>sinr</i>	<ul style="list-style-type: none"> <li>• SINR level</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>errorRateInfo</i>	<ul style="list-style-type: none"> <li>• error rate info</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>rsrqInfo</i>	<ul style="list-style-type: none"> <li>• rsrq info</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>lteSnrinfo</i>	<ul style="list-style-type: none"> <li>• lte Snr information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>lteRsrpinfo</i>	<ul style="list-style-type: none"> <li>• lte rsrp info</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>

### 8.316.2 Field Documentation

#### 8.316.2.1 nas\_ecioListElement nas\_SLQSSignalStrengthsInformation::ecioInfo

8.316.2.2 `nas_errorRateListElement` `nas_SLQSSignalStrengthsInformation::errorRateInfo`

8.316.2.3 `uint32_t` `nas_SLQSSignalStrengthsInformation::io`

8.316.2.4 `nas_lteRsrpInformation` `nas_SLQSSignalStrengthsInformation::lteRsrpInfo`

8.316.2.5 `nas_lteSnrInformation` `nas_SLQSSignalStrengthsInformation::lteSnrInfo`

8.316.2.6 `nas_rsrqInformation` `nas_SLQSSignalStrengthsInformation::rsrqInfo`

8.316.2.7 `nas_rxSignalStrengthListElement` `nas_SLQSSignalStrengthsInformation::rxSignalStrengthInfo`

8.316.2.8 `uint8_t` `nas_SLQSSignalStrengthsInformation::sinr`

## 8.317 `nas_SLQSSignalStrengthsTlv` Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- [nas\\_SLQSSignalStrengthsInformation](#) `sSLQSSignalStrengthsInfo`

### 8.317.1 Detailed Description

This structure contains parameters SLQS Signal Strength.

#### Parameters

<i>TlvPresent</i>	indicating the presence of the TLV in the QMI ind
<i>sSLQSSignalStrengthsInfo</i>	signal strength info

### 8.317.2 Field Documentation

8.317.2.1 `nas_SLQSSignalStrengthsInformation` `nas_SLQSSignalStrengthsTlv::sSLQSSignalStrengthsInfo`

8.317.2.2 `uint8_t` `nas_SLQSSignalStrengthsTlv::TlvPresent`

## 8.318 `nas_SrvDomainPrefTlv` Struct Reference

### Data Fields

- `uint32_t` [SrvDomainPref](#)
- `uint8_t` [TlvPresent](#)

### 8.318.1 Detailed Description

Contain the Service domain Preference for system selection preferences.



## Parameters

<i>SrvDomainPref</i>	<ul style="list-style-type: none"> <li>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>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

## 8.318.2 Field Documentation

8.318.2.1 uint32\_t nas\_SrvDomainPrefTlv::SrvDomainPref

8.318.2.2 uint8\_t nas\_SrvDomainPrefTlv::TlvPresent

## 8.319 nas\_SrvRegRestrictionTlv Struct Reference

## Data Fields

- uint32\_t [SrvRegRestriction](#)
- uint8\_t [TLVPresent](#)

## 8.319.1 Detailed Description

Structure for storing the Network Selection Registration Restriction

## Parameters

<i>SrvReg-Restriction</i>	<ul style="list-style-type: none"> <li>Registration restriction. <ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>* 0x00 - Device follows the normal registration process</li> <li>* 0x01 - Device follows the camp-only registration process</li> <li>* All other values are reserved.</li> </ul> </li> </ul> </li> </ul>
<i>TLVPresent</i>	<ul style="list-style-type: none"> <li>TLV Present</li> </ul>

## 8.319.2 Field Documentation

8.319.2.1 uint32\_t nas\_SrvRegRestrictionTlv::SrvRegRestriction

8.319.2.2 uint8\_t nas\_SrvRegRestrictionTlv::TLVPresent

## 8.320 nas\_SrvRegRestrictTlv Struct Reference

### Data Fields

- uint32\_t [SrvRegRestriction](#)
- uint8\_t [TlvPresent](#)

### 8.320.1 Detailed Description

Contain the Network Selection Registration Restriction Preference.

#### Parameters

<i>SrvReg-Restriction</i>	<ul style="list-style-type: none"> <li>• 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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

### 8.320.2 Field Documentation

8.320.2.1 uint32\_t nas\_SrvRegRestrictTlv::SrvRegRestriction

8.320.2.2 uint8\_t nas\_SrvRegRestrictTlv::TlvPresent

## 8.321 nas\_SrvStatusInfo Struct Reference

### Data Fields

- uint8\_t [srvStatus](#)
- uint8\_t [isPrefDataPath](#)

### 8.321.1 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>
-----------------------	---

### 8.321.2 Field Documentation

8.321.2.1 `uint8_t nas_SrvStatusInfo::isPrefDataPath`

8.321.2.2 `uint8_t nas_SrvStatusInfo::srvStatus`

## 8.322 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](#)

### 8.322.1 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>
-----------------------	--

### 8.322.2 Field Documentation

8.322.2.1 uint8\_t nas\_sysInfoCommon::isSysForbidden

8.322.2.2 uint8\_t nas\_sysInfoCommon::isSysForbiddenValid

8.322.2.3 uint8\_t nas\_sysInfoCommon::roamStatus

8.322.2.4 uint8\_t nas\_sysInfoCommon::roamStatusValid

8.322.2.5 uint8\_t nas\_sysInfoCommon::srvCapability

8.322.2.6 uint8\_t nas\_sysInfoCommon::srvCapabilityValid

8.322.2.7 uint8\_t nas\_sysInfoCommon::srvDomain

8.322.2.8 uint8\_t nas\_sysInfoCommon::srvDomainValid

## 8.323 nas\_TDSCDMABandPrefTlv Struct Reference

### Data Fields

- uint64\_t [TdsdmaBandPref](#)
- uint8\_t [TlvPresent](#)

### 8.323.1 Detailed Description

Contain the TDSCDMA Band Preference.

#### Parameters

<i>TdsdmaBand-Pref</i>	<ul style="list-style-type: none"> <li>Bitmask representing the TD-SCDMA band preference to be set.</li> <li>Values <ul style="list-style-type: none"> <li>0x01 - NAS_TDSCDMA_BAND_A - TD-SCDMA Band A</li> <li>0x02 - NAS_TDSCDMA_BAND_B - TD-SCDMA Band B</li> <li>0x04 - NAS_TDSCDMA_BAND_C - TD-SCDMA Band C</li> <li>0x08 - NAS_TDSCDMA_BAND_D - TD-SCDMA Band D</li> <li>0x10 - NAS_TDSCDMA_BAND_E - TD-SCDMA Band E</li> <li>0x20 - NAS_TDSCDMA_BAND_F - TD-SCDMA Band F</li> <li>All other bits are reserved.</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

### 8.323.2 Field Documentation

8.323.2.1 uint64\_t nas\_TDSCDMABandPrefTlv::TdscdmaBandPref

8.323.2.2 uint8\_t nas\_TDSCDMABandPrefTlv::TlvPresent

## 8.324 nas\_TDSCDMAECIOThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMAECIOThreshListLen](#)
- float \* [pTDSCDMAECIOThreshList](#)

### 8.324.1 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>

### 8.324.2 Field Documentation

8.324.2.1 float\* nas\_TDSCDMAECIOThresh::pTDSCDMAECIOThreshList

8.324.2.2 uint8\_t nas\_TDSCDMAECIOThresh::TDSCDMAECIOThreshListLen

## 8.325 nas\_TDSCDMARSCPThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMARSCPThreshListLen](#)
- int16\_t \* [pTDSCDMARSCPThreshList](#)

### 8.325.1 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>

### 8.325.2 Field Documentation

8.325.2.1 int16\_t\* nas\_TDSCDMARSCPThresh::pTDSCDMARSCPThreshList

8.325.2.2 uint8\_t nas\_TDSCDMARSCPThresh::TDSCDMARSCPThreshListLen

## 8.326 nas\_TDSCDMARSSIThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMARSSIThreshListLen](#)
- float \* [pTDSCDMARSSIThreshList](#)

### 8.326.1 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>

### 8.326.2 Field Documentation

8.326.2.1 float\* nas\_TDSCDMARSSIThresh::pTDSCDMARSSIThreshList

8.326.2.2 uint8\_t nas\_TDSCDMARSSIThresh::TDSCDMARSSIThreshListLen

## 8.327 nas\_TDSCDMASINRCONFThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMASINRCONFThreshListLen](#)
- float \* [pTDSCDMASINRCONFThreshList](#)

### 8.327.1 Detailed Description

This structure contains TDSCDMA SINR threshold list.

#### Parameters

<i>TDSCDMASINRCONFThreshListLen</i>	<ul style="list-style-type: none"> <li>• Length of the TDSCDMA SINR threshold list parameter to follow</li> </ul>
<i>pTDSCDMASINRCONFThreshList</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>

### 8.327.2 Field Documentation

8.327.2.1 float\* nas\_TDSCDMASINRCONFTthresh::pTDSCDMASINRCONFTthreshList

8.327.2.2 uint8\_t nas\_TDSCDMASINRCONFTthresh::TDSCDMASINRCONFTthreshListLen

## 8.328 nas\_TDSCDMASINRThresh Struct Reference

### Data Fields

- uint8\_t [TDSCDMASINRThreshListLen](#)
- float \* [pTDSCDMASINRThreshList](#)

### 8.328.1 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>

### 8.328.2 Field Documentation

8.328.2.1 float\* nas\_TDSCDMASINRThresh::pTDSCDMASINRThreshList

8.328.2.2 uint8\_t nas\_TDSCDMASINRThresh::TDSCDMASINRThreshListLen

## 8.329 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](#)

### 8.329.1 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>

### 8.329.2 Field Documentation

8.329.2.1 `uint8_t nas_timeInfo::day`

8.329.2.2 `uint8_t nas_timeInfo::dayLtSavingAdj`

8.329.2.3 `uint8_t nas_timeInfo::dayOfWeek`

8.329.2.4 `uint8_t nas_timeInfo::hour`

8.329.2.5 `uint8_t nas_timeInfo::minute`

8.329.2.6 `uint8_t nas_timeInfo::month`

8.329.2.7 `uint8_t nas_timeInfo::radioInterface`

8.329.2.8 `uint8_t nas_timeInfo::second`

8.329.2.9 `int8_t nas_timeInfo::timeZone`

8.329.2.10 `uint8_t nas_timeInfo::TlvPresent`

8.329.2.11 `uint16_t nas_timeInfo::year`

## 8.330 nas\_trueIMSI Struct Reference

### Data Fields

- `uint8_t mccT` [3]
- `uint16_t imsiT1112`
- `uint8_t imsiTS1` [7]
- `uint8_t imsiTS2` [3]
- `uint8_t imsiTaddrNum`

### 8.330.1 Detailed Description

This structure contains True IMSI.

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

### 8.330.2 Field Documentation

8.330.2.1 uint16\_t nas\_trueIMSI::imsiT1112

8.330.2.2 uint8\_t nas\_trueIMSI::imsiTaddrNum

8.330.2.3 uint8\_t nas\_trueIMSI::imsiTS1[7]

8.330.2.4 uint8\_t nas\_trueIMSI::imsiTS2[3]

8.330.2.5 uint8\_t nas\_trueIMSI::mccT[3]

## 8.331 nas\_txInfo Struct Reference

### Data Fields

- uint8\_t [isInTraffic](#)
- int32\_t [txPower](#)

### 8.331.1 Detailed Description

This structure contains tx radio 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>

### 8.331.2 Field Documentation

8.331.2.1 uint8\_t nas\_txInfo::isInTraffic

8.331.2.2 int32\_t nas\_txInfo::txPower

## 8.332 nas\_UMTSExtInfo 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](#)
- int16\_t [squal](#)
- int16\_t [srxLvl](#)
- uint8\_t [umtsInst](#)
- [nas\\_umtsInstArr](#) [umtsInstElement](#) [255]
- uint8\_t [geranInst](#)
- [nas\\_geranInstArr](#) [geranInstElement](#) [255]
- uint8\_t [TlvPresent](#)

### 8.332.1 Detailed Description

This structure contains information about the UMTS Extended Info.

#### 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>	<ul style="list-style-type: none"> <li>• MCC/MNC information</li> </ul>
<i>lac</i>	<ul style="list-style-type: none"> <li>• Location area code.</li> </ul>
<i>uarfcn</i>	<ul style="list-style-type: none"> <li>• UTRA absolute RF channel number.</li> </ul>
<i>psc</i>	<ul style="list-style-type: none"> <li>• Primary scrambling code.</li> </ul>
<i>rscp</i>	<ul style="list-style-type: none"> <li>• Received signal code power</li> <li>• The received power on one code measured in dBm on the primary CPICH channel of the serving cell.</li> </ul>
<i>ecio</i>	<ul style="list-style-type: none"> <li>• The received energy per chip divided by the power density in the band measured in dBm on the primary CPICH channel of the serving cell.</li> </ul>
<i>squal</i>	<ul style="list-style-type: none"> <li>• Cell selection quality value in dB.</li> </ul>
<i>srxLvl</i>	<ul style="list-style-type: none"> <li>• Cell selection Rx level value in dB.</li> </ul>
<i>umtsInst</i>	<ul style="list-style-type: none"> <li>• Number of sets of the <a href="#">umtsInstElement</a></li> </ul>

<i>umtsInstElement</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_umtsInstArr</a> for more information.</li> </ul>
<i>geranInst</i>	<ul style="list-style-type: none"> <li>• Number of sets of the <i>geranInstElement</i></li> </ul>
<i>geranInst-Element</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_geranInstArr</a> for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.332.2 Field Documentation

- 8.332.2.1 `uint16_t nas_UMTSExtInfo::cellId`
- 8.332.2.2 `int16_t nas_UMTSExtInfo::ecio`
- 8.332.2.3 `uint8_t nas_UMTSExtInfo::geranInst`
- 8.332.2.4 `nas_geranInstArr nas_UMTSExtInfo::geranInstElement[255]`
- 8.332.2.5 `uint16_t nas_UMTSExtInfo::lac`
- 8.332.2.6 `uint8_t nas_UMTSExtInfo::plmn[3]`
- 8.332.2.7 `uint16_t nas_UMTSExtInfo::psc`
- 8.332.2.8 `int16_t nas_UMTSExtInfo::rscp`
- 8.332.2.9 `int16_t nas_UMTSExtInfo::squal`
- 8.332.2.10 `int16_t nas_UMTSExtInfo::srxFvl`
- 8.332.2.11 `uint8_t nas_UMTSExtInfo::TlvPresent`
- 8.332.2.12 `uint16_t nas_UMTSExtInfo::uarfcn`
- 8.332.2.13 `uint8_t nas_UMTSExtInfo::umtsInst`
- 8.332.2.14 `nas_umtsInstArr nas_UMTSExtInfo::umtsInstElement[255]`

## 8.333 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]

### 8.333.1 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[NAS_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</i> [ <i>N-AS_MAX_DESCRIPTION_LENGTH</i> ]	<ul style="list-style-type: none"> <li>See <a href="#">nas_geranInstInfo</a> for more information.</li> </ul>
---	---

### 8.333.2 Field Documentation

- 8.333.2.1 `uint16_t nas_UMTSInfo::cellID`
- 8.333.2.2 `int16_t nas_UMTSInfo::ecio`
- 8.333.2.3 `uint8_t nas_UMTSInfo::geranInst`
- 8.333.2.4 `nas_geranInstInfo nas_UMTSInfo::GeranInstInfo[255]`
- 8.333.2.5 `uint16_t nas_UMTSInfo::lac`
- 8.333.2.6 `uint8_t nas_UMTSInfo::plmn[3]`
- 8.333.2.7 `uint16_t nas_UMTSInfo::psc`
- 8.333.2.8 `int16_t nas_UMTSInfo::rscp`
- 8.333.2.9 `uint16_t nas_UMTSInfo::uarfcn`
- 8.333.2.10 `uint8_t nas_UMTSInfo::umtsInst`
- 8.333.2.11 `nas_UMTSInstInfo nas_UMTSInfo::UMTSInstInfo[255]`

## 8.334 nas\_umtsInstArr Struct Reference

### Data Fields

- `uint16_t umtsUarfcn`
- `uint16_t umtsPsc`
- `int16_t umtsRscp`
- `int16_t umtsEcio`
- `int16_t umtsSqual`
- `int16_t umtsSrxLvl`
- `int16_t umtsRank`
- `uint8_t umtsSet`

### 8.334.1 Detailed Description

This structure contains information about the UMTS Info.

#### 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> <li>The received power on one code measured in dBm on the primary CPICH channel of the neighbor/monitored cell.</li> </ul>
<i>umtsEcio</i>	<ul style="list-style-type: none"> <li>ECIO</li> <li>The received energy per chip divided by the power density in the band measured in dBm on the primary CPICH channel of the neighbor/monitored cell.</li> </ul>
<i>umtsSqual</i>	<ul style="list-style-type: none"> <li>Squal</li> <li>cell selection quality value in dB.</li> </ul>
<i>umtsSrxLvl</i>	<ul style="list-style-type: none"> <li>Srxlev</li> <li>cell selection Rx level value in dB.</li> </ul>
<i>umtsRank</i>	<ul style="list-style-type: none"> <li>Rank of the cell.</li> </ul>
<i>umtsSet</i>	<ul style="list-style-type: none"> <li>Set of the cell.</li> </ul>

### 8.334.2 Field Documentation

8.334.2.1 int16\_t nas\_umtsInstArr::umtsEcio

8.334.2.2 uint16\_t nas\_umtsInstArr::umtsPsc

8.334.2.3 int16\_t nas\_umtsInstArr::umtsRank

8.334.2.4 int16\_t nas\_umtsInstArr::umtsRscp

8.334.2.5 uint8\_t nas\_umtsInstArr::umtsSet

8.334.2.6 int16\_t nas\_umtsInstArr::umtsSqual

8.334.2.7 int16\_t nas\_umtsInstArr::umtsSrxLvl

8.334.2.8 uint16\_t nas\_umtsInstArr::umtsUarfcn

## 8.335 nas\_UMTSinstInfo Struct Reference

### Data Fields

- uint16\_t [umtsUarfcn](#)
- uint16\_t [umtsPsc](#)
- int16\_t [umtsRscp](#)
- int16\_t [umtsEcio](#)



### 8.335.1 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>

### 8.335.2 Field Documentation

8.335.2.1 int16\_t nas\_UMTSinstInfo::umtsEcio

8.335.2.2 uint16\_t nas\_UMTSinstInfo::umtsPsc

8.335.2.3 int16\_t nas\_UMTSinstInfo::umtsRscp

8.335.2.4 uint16\_t nas\_UMTSinstInfo::umtsUarfcn

## 8.336 nas\_umtsLTENbrCell Struct Reference

### Data Fields

- uint16\_t [earfcn](#)
- uint16\_t [pci](#)
- uint32\_t [rsrp](#)
- uint32\_t [rsrq](#)
- int16\_t [srxlev](#)
- uint8\_t [cellsTDD](#)

### 8.336.1 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>

### 8.336.2 Field Documentation

8.336.2.1 uint8\_t nas\_umtsLTENbrCell::cellsTDD

8.336.2.2 uint16\_t nas\_umtsLTENbrCell::earfcn

8.336.2.3 uint16\_t nas\_umtsLTENbrCell::pci

8.336.2.4 uint32\_t nas\_umtsLTENbrCell::rsrp

8.336.2.5 uint32\_t nas\_umtsLTENbrCell::rsrq

8.336.2.6 int16\_t nas\_umtsLTENbrCell::srxlev

## 8.337 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](#)

### 8.337.1 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>

### 8.337.2 Field Documentation

8.337.2.1 uint8\_t nas\_UniversalTime::day

8.337.2.2 uint8\_t nas\_UniversalTime::dayOfWeek

8.337.2.3 uint8\_t nas\_UniversalTime::hour

8.337.2.4 uint8\_t nas\_UniversalTime::minute

8.337.2.5 uint8\_t nas\_UniversalTime::month

8.337.2.6 uint8\_t nas\_UniversalTime::second

8.337.2.7 uint16\_t nas\_UniversalTime::year

## 8.338 nas\_UsageSettingTlv Struct Reference

### Data Fields

- uint32\_t [UsageSetting](#)
- uint8\_t [TlvPresent](#)

### 8.338.1 Detailed Description

Contain the Usage Preference info.

## Parameters

<i>UsageSetting</i>	<ul style="list-style-type: none"> <li>• Modem usage preference to be set.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Unknown</li> <li>– 1 - Voice centric</li> <li>– 2 - Data centric</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.338.2 Field Documentation

8.338.2.1 uint8\_t nas\_UsageSettingTlv::TlvPresent

8.338.2.2 uint32\_t nas\_UsageSettingTlv::UsageSetting

## 8.339 nas\_VoiceDomainPrefTlv Struct Reference

## Data Fields

- uint32\_t [VoiceDomainPref](#)
- uint8\_t [TlvPresent](#)

## 8.339.1 Detailed Description

Contain the Voice Domain Preference Info.

## Parameters

<i>VoiceDomain-Pref</i>	<ul style="list-style-type: none"> <li>• Voice domain preference.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Circuit-switched (CS) voice only</li> <li>– 0x01 - Packet-switched (PS) voice only</li> <li>– 0x02 - CS is preferred; PS is secondary</li> <li>– 0x03 - PS is preferred; CS is secondary</li> </ul> </li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present</li> </ul>

## 8.339.2 Field Documentation

8.339.2.1 uint8\_t nas\_VoiceDomainPrefTlv::TlvPresent

8.339.2.2 uint32\_t nas\_VoiceDomainPrefTlv::VoiceDomainPref

## 8.340 nas\_wcdmaCellInfo Struct Reference

## Data Fields

- uint16\_t [psc](#)
- int16\_t [cpich\\_rscp](#)
- int16\_t [cpich\\_ecno](#)
- int16\_t [srxlev](#)

## 8.340.1 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>

## 8.340.2 Field Documentation

8.340.2.1 int16\_t nas\_wcdmaCellInfo::cpich\_ecno

8.340.2.2 int16\_t nas\_wcdmaCellInfo::cpich\_rscp

8.340.2.3 uint16\_t nas\_wcdmaCellInfo::psc

8.340.2.4 int16\_t nas\_wcdmaCellInfo::srxlev

## 8.341 nas\_WCDMACellInfoExt Struct Reference

## Data Fields

- float [wAgc](#)
- float [wTxAgc](#)
- uint16\_t [wDIBler](#)
- uint8\_t [TlvPresent](#)

### 8.341.1 Detailed Description

This structure contains information about the WCDMA Cell Info Ext.

#### Parameters

<i>wAgc</i>	<ul style="list-style-type: none"> <li>Power in dB</li> </ul>
<i>wTxAgc</i>	<ul style="list-style-type: none"> <li>Tx power in dB.</li> </ul>
<i>wDIBler</i>	<ul style="list-style-type: none"> <li>Downlink block error rate percentage.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present</li> </ul>

### 8.341.2 Field Documentation

8.341.2.1 `uint8_t nas_WCDMACellInfoExt::TlvPresent`

8.341.2.2 `float nas_WCDMACellInfoExt::wAgc`

8.341.2.3 `uint16_t nas_WCDMACellInfoExt::wDIBler`

8.341.2.4 `float nas_WCDMACellInfoExt::wTxAgc`

## 8.342 nas\_WCDMAECIOThresh Struct Reference

### Data Fields

- `uint8_t` [WCDMAECIOThreshListLen](#)
- `int16_t *` [pWCDMAECIOThreshList](#)

### 8.342.1 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>

### 8.342.2 Field Documentation

8.342.2.1 int16\_t\* nas\_WCDMAECIOThresh::pWCDMAECIOThreshList

8.342.2.2 uint8\_t nas\_WCDMAECIOThresh::WCDMAECIOThreshListLen

## 8.343 nas\_WCDMAInfoLTENeighborCell Struct Reference

### Data Fields

- uint32\_t [wcdmaRRCTest](#)
- uint8\_t [umtsLTENbrCellLen](#)
- [nas\\_umtsLTENbrCell](#) UMTSLTENbrCell [255]

### 8.343.1 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>

### 8.343.2 Field Documentation

8.343.2.1 nas\_umtsLTENbrCell nas\_WCDMAInfoLTENeighborCell::UMTSLTENbrCell[255]

8.343.2.2 uint8\_t nas\_WCDMAInfoLTENeighborCell::umtsLTENbrCellLen

8.343.2.3 uint32\_t nas\_WCDMAInfoLTENeighborCell::wcdmaRRCTest

## 8.344 nas\_WCDMARSSIThresh Struct Reference

## Data Fields

- uint8\_t [WCDMARSSIThreshListLen](#)
- int16\_t \* [pWCDMARSSIThreshList](#)

### 8.344.1 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>

### 8.344.2 Field Documentation

8.344.2.1 int16\_t\* nas\_WCDMARSSIThresh::pWCDMARSSIThreshList

8.344.2.2 uint8\_t nas\_WCDMARSSIThresh::WCDMARSSIThreshListLen

## 8.345 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](#)

### 8.345.1 Detailed Description

Structure for storing the WCDMA System Information.



## Parameters

<i>sysInfoWCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_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[NAS_PLM-N_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Mobile Country Code.</li> <li>• MCC digits in ASCII characters</li> </ul>
<i>MNC[NAS_PLM-N_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>

## 8.345.2 Field Documentation

- 8.345.2.1 `uint32_t nas_WCDMASysInfo::cellId`
- 8.345.2.2 `uint8_t nas_WCDMASysInfo::cellIdValid`
- 8.345.2.3 `uint8_t nas_WCDMASysInfo::hsCallStatus`
- 8.345.2.4 `uint8_t nas_WCDMASysInfo::hsCallStatusValid`
- 8.345.2.5 `uint8_t nas_WCDMASysInfo::hsInd`
- 8.345.2.6 `uint8_t nas_WCDMASysInfo::hsIndValid`
- 8.345.2.7 `uint16_t nas_WCDMASysInfo::lac`
- 8.345.2.8 `uint8_t nas_WCDMASysInfo::lacValid`
- 8.345.2.9 `uint8_t nas_WCDMASysInfo::MCC[3]`
- 8.345.2.10 `uint8_t nas_WCDMASysInfo::MNC[3]`
- 8.345.2.11 `uint8_t nas_WCDMASysInfo::networkIdValid`
- 8.345.2.12 `uint16_t nas_WCDMASysInfo::psc`
- 8.345.2.13 `uint8_t nas_WCDMASysInfo::pscValid`
- 8.345.2.14 `uint8_t nas_WCDMASysInfo::regRejectInfoValid`
- 8.345.2.15 `uint8_t nas_WCDMASysInfo::rejCause`
- 8.345.2.16 `uint8_t nas_WCDMASysInfo::rejectSrvDomain`
- 8.345.2.17 `nas_sysInfoCommon nas_WCDMASysInfo::sysInfoWCDMA`

## 8.346 nas\_wcdmaUARFCN Struct Reference

### Data Fields

- `uint8_t` [status](#)
- `uint32_t` [uarfcn](#)

### 8.346.1 Detailed Description

This structure contains WCDMA UARFCN information.

#### 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>

### 8.346.2 Field Documentation

8.346.2.1 `uint8_t nas_wcdmaUARFCN::status`

8.346.2.2 `uint32_t nas_wcdmaUARFCN::uarfcn`

## 8.347 NASAcqOrderPrefTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t AcqOrderLen`
- `uint8_t AcqOrderPref` [255]

### 8.347.1 Detailed Description

Structure used to store acquisition order preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>AcqOrderPref</i>	- Acquisition order Preference Value

### 8.347.2 Field Documentation

8.347.2.1 `uint8_t NASAcqOrderPrefTlv::AcqOrderLen`

8.347.2.2 `uint8_t NASAcqOrderPrefTlv::AcqOrderPref`[255]

8.347.2.3 `uint8_t NASAcqOrderPrefTlv::TlvPresent`

## 8.348 NASBandPreferenceTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t band_pref`

### 8.348.1 Detailed Description

Structure used to store all Band Preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>band_pref</i>	- Roaming Indication Value

### 8.348.2 Field Documentation

8.348.2.1 `uint64_t NASBandPreferenceTlv::band_pref`

8.348.2.2 `uint8_t NASBandPreferenceTlv::TlvPresent`

## 8.349 NASCiotAcqOrderPrefTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t CiotAcqOrderLen`
- `uint32_t CiotAcqOrderPref [255]`

### 8.349.1 Detailed Description

Structure used to store CIOT acquisition order preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>CiotAcqOrder-Pref</i>	- CIOT Acquisition order Preference Value

### 8.349.2 Field Documentation

8.349.2.1 `uint8_t NASCiotAcqOrderPrefTlv::CiotAcqOrderLen`

8.349.2.2 `uint32_t NASCiotAcqOrderPrefTlv::CiotAcqOrderPref[255]`

8.349.2.3 `uint8_t NASCiotAcqOrderPrefTlv::TlvPresent`

## 8.350 NASCiotLteOpModePrefTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t CiotLteOpModePref`

### 8.350.1 Detailed Description

Structure used to store CIOT LTE Operational Mode Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>CiotLteOpMode-Pref</i>	- Bitmask representing the CIoT LTE operational mode preference

## 8.350.2 Field Documentation

8.350.2.1 uint64\_t NASCiotLteOpModePrefTlv::CiotLteOpModePref

8.350.2.2 uint8\_t NASCiotLteOpModePrefTlv::TlvPresent

## 8.351 NASEmergencyModeTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [EmerMode](#)

## 8.351.1 Detailed Description

Structure used to store Emergency Mode TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>EmerMode</i>	- Emergency Mode Value

## 8.351.2 Field Documentation

8.351.2.1 uint8\_t NASEmergencyModeTlv::EmerMode

8.351.2.2 uint8\_t NASEmergencyModeTlv::TlvPresent

## 8.352 NasGetLTECphyCaInfo Struct Reference

## Data Fields

- [NASPhyCaAggScellIndType](#) [PhyCaAggScellIndType](#)
- [NASPhyCaAggScellDIBw](#) [PhyCaAggScellDIBw](#)
- [NASPhyCaAggScellInfo](#) [PhyCaAggScellInfo](#)
- [NASPhyCaAggPcellInfo](#) [PhyCaAggPcellInfo](#)
- [NASPhyCaAggScellIndex](#) [PhyCaAggScellIndex](#)
- [NASPhyCaAggScellArray](#) [PhyCaAggScellArray](#)

## 8.352.1 Detailed Description

This structure contains the parameters for aggregation event information.

## Parameters

<i>PhyCaAggScell-IndType</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggScellIndType</a></li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>PhyCaAggScell-DIBw</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggScellDIBw</a></li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>PhyCaAggScell-Info</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggScellInfo</a></li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>PhyCaAggPcell-Info</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggPcellInfo</a></li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>PhyCaAggScell-Index</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggScellIndex</a></li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>PhyCaAggScell-Array</i>	<ul style="list-style-type: none"> <li>• see <a href="#">NASPhyCaAggScellArray</a></li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>

## 8.352.2 Field Documentation

8.352.2.1 **NASPhyCaAggPcellInfo** NasGetLTECphyCalInfo::PhyCaAggPcellInfo8.352.2.2 **NASPhyCaAggScellArray** NasGetLTECphyCalInfo::PhyCaAggScellArray8.352.2.3 **NASPhyCaAggScellDIBw** NasGetLTECphyCalInfo::PhyCaAggScellDIBw8.352.2.4 **NASPhyCaAggScellIndex** NasGetLTECphyCalInfo::PhyCaAggScellIndex8.352.2.5 **NASPhyCaAggScellIndType** NasGetLTECphyCalInfo::PhyCaAggScellIndType8.352.2.6 **NASPhyCaAggScellInfo** NasGetLTECphyCalInfo::PhyCaAggScellInfo

## 8.353 NASGWAcqOrderPrefTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [GWAcqOrderPref](#)

## 8.353.1 Detailed Description

Structure used to store GSM/WCDMA acquisition order preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>GWAcqOrder-Pref</i>	- GSM/WCDMA acquisition order Preference Value

## 8.353.2 Field Documentation

8.353.2.1 uint32\_t NASGWAcqOrderPrefTlv::GWAcqOrderPref

8.353.2.2 uint8\_t NASGWAcqOrderPrefTlv::TlvPresent

## 8.354 NASLTEBandPreferenceTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [LTEBandPref](#)

## 8.354.1 Detailed Description

Structure used to store LTE Band Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>LTEBandPref</i>	- LTE Band Preference Value

## 8.354.2 Field Documentation

8.354.2.1 uint64\_t NASLTEBandPreferenceTlv::LTEBandPref

8.354.2.2 uint8\_t NASLTEBandPreferenceTlv::TlvPresent

## 8.355 NASLteM1BandPrefTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [LteM1BandPref](#)

## 8.355.1 Detailed Description

Purpose: Structure used to store LTE M1 Band Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>LteM1BandPref</i>	- Bitmask representing the LTE M1 band preference

## 8.355.2 Field Documentation



8.355.2.1 `uint64_t NASLteM1BandPrefTlv::LteM1BandPref`

8.355.2.2 `uint8_t NASLteM1BandPrefTlv::TlvPresent`

## 8.356 NASLteNasReleaseInfoTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t nas_release`
- `uint8_t nas_major`
- `uint8_t nas_minor`

### 8.356.1 Detailed Description

This structure contains LTE Nas Release Information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>
<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>

### 8.356.2 Field Documentation

8.356.2.1 `uint8_t NASLteNasReleaseInfoTlv::nas_major`

8.356.2.2 `uint8_t NASLteNasReleaseInfoTlv::nas_minor`

8.356.2.3 `uint8_t NASLteNasReleaseInfoTlv::nas_release`

8.356.2.4 `uint8_t NASLteNasReleaseInfoTlv::TlvPresent`

## 8.357 NASLteNB1BandPrefTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t LteNB1BandPref`

### 8.357.1 Detailed Description

Structure used to store LTE NB1 Band Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>LteNB1BandPref</i>	- Bitmask representing the LTE NB1 band preference

## 8.357.2 Field Documentation

8.357.2.1 uint64\_t NASLteNB1BandPrefTlv::LteNB1BandPref

8.357.2.2 uint8\_t NASLteNB1BandPrefTlv::TlvPresent

## 8.358 NASModePreferenceTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [ModePref](#)

## 8.358.1 Detailed Description

Structure used to store Mode Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>ModePref</i>	- Mode Preference Value

## 8.358.2 Field Documentation

8.358.2.1 uint16\_t NASModePreferenceTlv::ModePref

8.358.2.2 uint8\_t NASModePreferenceTlv::TlvPresent

## 8.359 NASNetSelPreferenceTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [NetSelPref](#)

## 8.359.1 Detailed Description

Structure used to store Network Selection Preference TLV Value.

## Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>NetSelPref</i>	- Network Selection Preference Value

## 8.359.2 Field Documentation

8.359.2.1 uint8\_t NASNetSelPreferenceTlv::NetSelPref

8.359.2.2 `uint8_t NASNetSelPreferenceTlv::TlvPresent`

## 8.360 NASNr5gBandPrefTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t Nr5gBandPrefbits1_64`
- `uint64_t Nr5gBandPrefbits_65_128`
- `uint64_t Nr5gBandPrefbits_129_192`
- `uint64_t Nr5gBandPrefbits_193_256`

### 8.360.1 Detailed Description

Structure used to store NR5G Band Preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>Nr5gBandPref</i>	- NR5G Band Preference

### 8.360.2 Field Documentation

8.360.2.1 `uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits1_64`

8.360.2.2 `uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_129_192`

8.360.2.3 `uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_193_256`

8.360.2.4 `uint64_t NASNr5gBandPrefTlv::Nr5gBandPrefbits_65_128`

8.360.2.5 `uint8_t NASNr5gBandPrefTlv::TlvPresent`

## 8.361 NASOTAMessageTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t message_type`
- `uint16_t data_len`
- `uint8_t data_buf` [2048]

### 8.361.1 Detailed Description

This structure contains OTA message

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• unpack result</li></ul>
------------------	---

<i>message_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_buf</i>	<ul style="list-style-type: none"> <li>• OTA Message Content</li> </ul>

## 8.361.2 Field Documentation

8.361.2.1 `uint8_t NASOTAMessageTlv::data_buf[2048]`

8.361.2.2 `uint16_t NASOTAMessageTlv::data_len`

8.361.2.3 `uint32_t NASOTAMessageTlv::message_type`

8.361.2.4 `uint8_t NASOTAMessageTlv::TlvPresent`

## 8.362 NASPhyCaAggPcellInfo Struct Reference

### Data Fields

- `uint32_t pci`
- `uint32_t freq`
- `LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value`
- `uint32_t iLTEbandValue`
- `uint8_t TlvPresent`

### 8.362.1 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="#">LITEQMI_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="#">LITEQMI_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>

### 8.362.2 Field Documentation

8.362.2.1 [LITEQMI\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) NASPhyCaAggPcellInfo::dl\_bw\_value

8.362.2.2 [uint32\\_t](#) NASPhyCaAggPcellInfo::freq

8.362.2.3 [uint32\\_t](#) NASPhyCaAggPcellInfo::iLTEbandValue

8.362.2.4 [uint32\\_t](#) NASPhyCaAggPcellInfo::pci

8.362.2.5 [uint8\\_t](#) NASPhyCaAggPcellInfo::TlvPresent

## 8.363 NASPhyCaAggScellArray Struct Reference

### Data Fields

- [uint8\\_t](#) [cphy\\_scell\\_info\\_list\\_len](#)
- [uint16\\_t](#) [pci](#) [255]
- [uint16\\_t](#) [freq](#) [255]
- [LITEQMI\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) [cphy\\_ca\\_dl\\_bandwidth](#) [255]
- [uint16\\_t](#) [band](#) [255]
- [LITEQMI\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) [scell\\_state](#) [255]
- [uint16\\_t](#) [scell\\_idx](#) [255]
- [uint8\\_t](#) [TlvPresent](#)

### 8.363.1 Detailed Description

This structure contains the parameters for Physical Carrier aggregation of Scell Index.

#### Parameters

<i>cphy_scell_info_list_len</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements: <ul style="list-style-type: none"> <li><a href="#">pci</a></li> <li><a href="#">freq</a></li> <li><a href="#">cphy_ca_dl_bandwidth</a></li> <li><a href="#">band</a></li> <li><a href="#">scell_state</a></li> <li><a href="#">scell_idx</a></li> </ul> </li> </ul>
---------------------------------	--

<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>cphy_ca_dl_bandwidth</i>	<ul style="list-style-type: none"> <li>Downlink Bandwidth Values.</li> <li>See <a href="#">LITEQMI_NAS_LTE_CPHY_CA_BW_NRB</a> for more information.</li> </ul>
<i>band</i>	<ul style="list-style-type: none"> <li>Band value.</li> <li>See <a href="#">LITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND</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="#">LITEQMI_NAS_LTE_CPHY_SCELL_STATE</a> for more information.</li> </ul>
<i>scell_idx</i>	<ul style="list-style-type: none"> <li>Scell index Values.</li> <li>Range for index values: 0 to 7.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

### 8.363.2 Field Documentation

8.363.2.1 `uint16_t NASPhyCaAggScellArray::band[255]`

8.363.2.2 `LITEQMI_NAS_LTE_CPHY_CA_BW_NRB NASPhyCaAggScellArray::cphy_ca_dl_bandwidth[255]`

8.363.2.3 `uint8_t NASPhyCaAggScellArray::cphy_scell_info_list_len`

8.363.2.4 `uint16_t NASPhyCaAggScellArray::freq[255]`

8.363.2.5 `uint16_t NASPhyCaAggScellArray::pci[255]`

8.363.2.6 `uint16_t NASPhyCaAggScellArray::scell_idx[255]`

8.363.2.7 `LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellArray::scell_state[255]`

8.363.2.8 `uint8_t NASPhyCaAggScellArray::TlvPresent`

## 8.364 NASPhyCaAggScellIDBw Struct Reference

### Data Fields

- [LITEQMI\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB dl\\_bw\\_value](#)
- `uint8_t` [TlvPresent](#)

### 8.364.1 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="#">LITEQMI_NAS_LTE_CPHY_CA_BW_NRB</a> for more information.</li> </ul>
--------------------	--

### 8.364.2 Field Documentation

8.364.2.1 [LITEQMI\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) NASPhyCaAggScellIDBw::dl\_bw\_value

8.364.2.2 [uint8\\_t](#) NASPhyCaAggScellIDBw::TlvPresent

## 8.365 NASPhyCaAggScellIndex Struct Reference

### Data Fields

- [uint8\\_t](#) [scell\\_idx](#)
- [uint8\\_t](#) [TlvPresent](#)

### 8.365.1 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>

### 8.365.2 Field Documentation

8.365.2.1 [uint8\\_t](#) NASPhyCaAggScellIndex::scell\_idx

8.365.2.2 [uint8\\_t](#) NASPhyCaAggScellIndex::TlvPresent

## 8.366 NASPhyCaAggScellIndType Struct Reference

### Data Fields

- [uint32\\_t](#) [pci](#)
- [uint32\\_t](#) [freq](#)
- [LITEQMI\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) [scell\\_state](#)
- [uint8\\_t](#) [TlvPresent](#)

### 8.366.1 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>cell_state</i>	<ul style="list-style-type: none"> <li>Scell state Values.</li> <li>See <a href="#">LITEQMI_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>

### 8.366.2 Field Documentation

8.366.2.1 `uint32_t NASPhyCaAggScellIndType::freq`

8.366.2.2 `uint32_t NASPhyCaAggScellIndType::pci`

8.366.2.3 `LITEQMI_NAS_LTE_CPHY_SCELL_STATE NASPhyCaAggScellIndType::cell_state`

8.366.2.4 `uint8_t NASPhyCaAggScellIndType::TlvPresent`

## 8.367 NASPhyCaAggScellInfo Struct Reference

#### Data Fields

- `uint32_t pci`
- `uint32_t freq`
- `LITEQMI_NAS_LTE_CPHY_CA_BW_NRB dl_bw_value`
- `uint32_t iLTEbandValue`
- `LITEQMI_NAS_LTE_CPHY_SCELL_STATE cell_state`
- `uint8_t TlvPresent`

### 8.367.1 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="#">LITEQMI_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="#">LITEQMI_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>

## 8.367.2 Field Documentation

8.367.2.1 [LITEQMI\\_NAS\\_LTE\\_CPHY\\_CA\\_BW\\_NRB](#) NASPhyCaAggScellInfo::dl\_bw\_value

8.367.2.2 [uint32\\_t](#) NASPhyCaAggScellInfo::freq

8.367.2.3 [uint32\\_t](#) NASPhyCaAggScellInfo::iLTEbandValue

8.367.2.4 [uint32\\_t](#) NASPhyCaAggScellInfo::pci

8.367.2.5 [LITEQMI\\_NAS\\_LTE\\_CPHY\\_SCELL\\_STATE](#) NASPhyCaAggScellInfo::scell\_state

8.367.2.6 [uint8\\_t](#) NASPhyCaAggScellInfo::TlvPresent

## 8.368 NASPRLPreferenceTlv Struct Reference

### Data Fields

- [uint8\\_t](#) [TlvPresent](#)
- [uint16\\_t](#) [PRLPref](#)

### 8.368.1 Detailed Description

Structure used to store CDMA PRL Preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>PRLPref</i>	- CDMA PRL Preference Value

## 8.368.2 Field Documentation

8.368.2.1 uint16\_t NASPRLPreferenceTlv::PRLPref

8.368.2.2 uint8\_t NASPRLPreferenceTlv::TlvPresent

## 8.369 NASQmiCbkNasSwiOTAMessageInd Struct Reference

### Data Fields

- [NASOTAMessageTlv otaMsgTlv](#)
- [NASLteNasReleaseInfoTlv nasRelInfoTlv](#)
- [NASTimeInfoTlv timeTlv](#)

### 8.369.1 Detailed Description

This structure contains OTA message

#### Parameters

<i>otaMsgTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">NASOTAMessageTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>nasRelInfoTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">NASLteNasReleaseInfoTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>timeTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">NASTimeInfoTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

### 8.369.2 Field Documentation

8.369.2.1 NASLteNasReleaseInfoTlv NASQmiCbkNasSwiOTAMessageInd::nasRelInfoTlv

8.369.2.2 NASOTAMessageTlv NASQmiCbkNasSwiOTAMessageInd::otaMsgTlv

8.369.2.3 NASTimeInfoTlv NASQmiCbkNasSwiOTAMessageInd::timeTlv

## 8.370 NASQmiCbkNasSystemSelPrefInd Struct Reference

### Data Fields

- [NASEmergencyModeTlv EMTlv](#)
- [NASModePreferenceTlv MPTlv](#)
- [NASBandPreferenceTlv BPTlv](#)
- [NASPRLPreferenceTlv PRLPTlv](#)
- [NASRoamPreferenceTlv RPTlv](#)
- [NASLTEBandPreferenceTlv LBPTlv](#)
- [NASNetSelPreferenceTlv NSPTlv](#)
- [NASServDomainPrefTlv SDPTlv](#)
- [NASGWAcqOrderPrefTlv GWAOPTlv](#)
- [NASAcqOrderPrefTlv AOPTlv](#)

- [NASRatDisabledMaskTlv](#) RatDMTlv
- [NASCiotLteOpModePrefTlv](#) CiotOpMPTlv
- [NASLteM1BandPrefTlv](#) M1BandPTlv
- [NASLteNB1BandPrefTlv](#) NB1BandPTlv
- [NASCiotAcqOrderPrefTlv](#) CiotAOPTlv
- [NASNr5gBandPrefTlv](#) NR5gBandPTlv

### 8.370.1 Detailed Description

Structure used to store all QMI Notification parameters.

#### Parameters

<i>EMTlv</i>	- Emergency mode Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASEmergencyModeTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>MPTlv</i>	- Mode preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASModePreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>BPTlv</i>	- Band preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASBandPreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>PRLPTlv</i>	- PRL preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASPRLPreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>RPTlv</i>	- Roaming preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASRoamPreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>LBPTlv</i>	- LTE band preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASLTEBandPreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>NSPTlv</i>	- Network selection preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASNetSelPreferenceTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>SDPTlv</i>	- Service domain preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASServDomainPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>GWAOPTlv</i>	- GSM WCDMA Acquisition Order Preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASGWAcqOrderPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>AOPTlv</i>	- Acquisition Order Preference Tlv <ul style="list-style-type: none"> <li>• See <a href="#">NASAcqOrderPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>

<i>RatDMTlv</i>	<ul style="list-style-type: none"> <li>- Disabled RAT Bitmask Tlv</li> <li>• See <a href="#">NASRatDisabledMaskTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>CiotOpMPTlv</i>	<ul style="list-style-type: none"> <li>- CIOT LTE Operational Mode Preference Tlv</li> <li>• See <a href="#">NASCiotLteOpModePrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>M1BandPTlv</i>	<ul style="list-style-type: none"> <li>- LTE M1 Band Preference Tlv</li> <li>• See <a href="#">NASLteM1BandPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>NB1BandPTlv</i>	<ul style="list-style-type: none"> <li>- LTE NB1 Band Preference Tlv</li> <li>• See <a href="#">NASLteNB1BandPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>CiotAOPTlv</i>	<ul style="list-style-type: none"> <li>- CIOT acquisition order preference Tlv</li> <li>• See <a href="#">NASCiotAcqOrderPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>NR5gBandPTlv</i>	<ul style="list-style-type: none"> <li>- NR5G Band Preference Tlv</li> <li>• See <a href="#">NASNr5gBandPrefTlv</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>

Note: None

## 8.370.2 Field Documentation

- 8.370.2.1 **NASAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::AOPTlv
- 8.370.2.2 **NASBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::BPTlv
- 8.370.2.3 **NASCiotAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::CiotAOPTlv
- 8.370.2.4 **NASCiotLteOpModePrefTlv** NASQmiCbkNasSystemSelPrefInd::CiotOpMPTlv
- 8.370.2.5 **NASEmergencyModeTlv** NASQmiCbkNasSystemSelPrefInd::EMTlv
- 8.370.2.6 **NASGWAcqOrderPrefTlv** NASQmiCbkNasSystemSelPrefInd::GWAOPTlv
- 8.370.2.7 **NASLTEBandPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::LBPTlv
- 8.370.2.8 **NASLteM1BandPrefTlv** NASQmiCbkNasSystemSelPrefInd::M1BandPTlv
- 8.370.2.9 **NASModePreferenceTlv** NASQmiCbkNasSystemSelPrefInd::MPTlv
- 8.370.2.10 **NASLteNB1BandPrefTlv** NASQmiCbkNasSystemSelPrefInd::NB1BandPTlv
- 8.370.2.11 **NASNr5gBandPrefTlv** NASQmiCbkNasSystemSelPrefInd::NR5gBandPTlv
- 8.370.2.12 **NASNetSelPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::NSPTlv

8.370.2.13 **NASPRLLPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::PRLPTlv

8.370.2.14 **NASRatDisabledMaskTlv** NASQmiCbkNasSystemSelPrefInd::RatDMTlv

8.370.2.15 **NASRoamPreferenceTlv** NASQmiCbkNasSystemSelPrefInd::RPTlv

8.370.2.16 **NASServDomainPrefTlv** NASQmiCbkNasSystemSelPrefInd::SDPTlv

## 8.371 NASRatDisabledMaskTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [RatDisabledMask](#)

### 8.371.1 Detailed Description

Structure used to store Disabled RAT Bitmask TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>RatDisabled-Mask</i>	- Bitmask representing the radio technologies that are disabled

### 8.371.2 Field Documentation

8.371.2.1 uint16\_t NASRatDisabledMaskTlv::RatDisabledMask

8.371.2.2 uint8\_t NASRatDisabledMaskTlv::TlvPresent

## 8.372 NASRoamPreferenceTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [RoamPref](#)

### 8.372.1 Detailed Description

Structure used to store Roaming Preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>RoamPref</i>	- Roaming Preference Value

### 8.372.2 Field Documentation

8.372.2.1 uint16\_t NASRoamPreferenceTlv::RoamPref

8.372.2.2 uint8\_t NASRoamPreferenceTlv::TlvPresent

## 8.373 NASServDomainPrefTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [SrvDomainPref](#)

### 8.373.1 Detailed Description

Structure used to store Service domain preference TLV Value.

#### Parameters

<i>TlvPresent</i>	- Boolean indicating the presence of the TLV in the QMI response
<i>SrvDomainPref</i>	- Service Domain Preference Value

### 8.373.2 Field Documentation

8.373.2.1 uint32\_t NASServDomainPrefTlv::SrvDomainPref

8.373.2.2 uint8\_t NASServDomainPrefTlv::TlvPresent

## 8.374 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](#)

### 8.374.1 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

## 8.374.2 Field Documentation

8.374.2.1 uint8\_t NAServingSystemInfo::csAttachState

8.374.2.2 uint8\_t NAServingSystemInfo::hdrPersonality

8.374.2.3 uint8\_t NAServingSystemInfo::psAttachState

8.374.2.4 uint8\_t NAServingSystemInfo::radioInterfaceList[255]

8.374.2.5 uint8\_t NAServingSystemInfo::radioInterfaceNo

8.374.2.6 uint8\_t NAServingSystemInfo::registrationState

8.374.2.7 uint8\_t NAServingSystemInfo::selectedNetwork

## 8.375 NASTimeInfoTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [time](#)

### 8.375.1 Detailed Description

This structure contains LTE Nas time Infomation

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>
<i>time</i>	<ul style="list-style-type: none"> <li>Seconds in local time since Jan. 6th 1980 00:00:00 UTC</li> </ul>

### 8.375.2 Field Documentation

8.375.2.1 `uint64_t` `NASTimeInfoTlv::time`

8.375.2.2 `uint8_t` `NASTimeInfoTlv::TlvPresent`

## 8.376 newMTMessageTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- [sMSMTMessageInfo](#) `MTMessageInfo`

### 8.376.1 Detailed Description

This structure contains MT message TLV information.

#### 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>

### 8.376.2 Field Documentation

8.376.2.1 `sMSMTMessageInfo` `newMTMessageTlv::MTMessageInfo`

8.376.2.2 `uint8_t` `newMTMessageTlv::TlvPresent`

## 8.377 pack\_audio\_SLQSGetAudioPathConfig\_t Struct Reference

### Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [Item](#)



### 8.377.1 Detailed Description

This structure contains the pack parameters for Get Audio Path Config.

#### 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_CODECASTG</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>

### 8.377.2 Field Documentation

8.377.2.1 uint8\_t pack\_audio\_SLQSGetAudioPathConfig\_t::Item

8.377.2.2 uint8\_t pack\_audio\_SLQSGetAudioPathConfig\_t::Profile

## 8.378 pack\_audio\_SLQSGetAudioProfile\_t Struct Reference

### Data Fields

- uint8\_t [Generator](#)

### 8.378.1 Detailed Description

This structure contains the pack parameters to Get Audio Profile.

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

### 8.378.2 Field Documentation

8.378.2.1 `uint8_t pack_audio_SLQSGetAudioProfile_t::Generator`

## 8.379 `pack_audio_SLQSGetAudioVoITLBConfig_t` Struct Reference

### Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [Generator](#)
- `uint8_t` [Volume](#)
- `uint8_t` [Item](#)

### 8.379.1 Detailed Description

This structure contains the pack parameters for `SLQSGetAudioVoITLBConfig`.

#### 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>

### 8.379.2 Field Documentation

8.379.2.1 `uint8_t pack_audio_SLQSGetAudioVoITLBConfig_t::Generator`

8.379.2.2 `uint8_t pack_audio_SLQSGetAudioVoITLBConfig_t::Item`

8.379.2.3 `uint8_t pack_audio_SLQSGetAudioVoITLBConfig_t::Profile`

8.379.2.4 `uint8_t pack_audio_SLQSGetAudioVoITLBConfig_t::Volume`

## 8.380 `pack_audio_SLQSSetAudioPathConfig_t` Struct Reference

### Data Fields

- `uint8_t` [Profile](#)

- uint8\_t \* [pECMode](#)
- uint8\_t \* [pNSEnable](#)
- uint16\_t \* [pTXGain](#)
- uint16\_t \* [pDTMFTXGain](#)
- uint16\_t \* [pCodecSTGain](#)
- [audio\\_TXPCMIIRFiltr](#) \* [pTXPCMIIRFiltr](#)
- [audio\\_RXPCMIIRFiltr](#) \* [pRXPCMIIRFiltr](#)
- uint8\_t \* [pRXAVCAGCSwitch](#)
- uint8\_t \* [pTXAVCSwitch](#)
- [audio\\_RXAGCList](#) \* [pRXAGCList](#)
- [audio\\_RXAVCList](#) \* [pRXAVCList](#)
- [audio\\_TXAGCList](#) \* [pTXAGCList](#)

### 8.380.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioPathConfig .

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

<i>pRXAVCAGC-Switch</i>	[Optional] • RX AVC/AGC Switch
<i>pTXAVCSwitch</i>	[Optional] • TX AVC Switch
<i>pRXAGCList</i>	[Optional] • See <a href="#">audio_RXAGCList</a> for more information
<i>pRXAVCList</i>	[Optional] • See <a href="#">audio_RXAVCList</a> for more information
<i>pTXAGCList</i>	[Optional] • See <a href="#">audio_TXAGCList</a> for more information

## 8.380.2 Field Documentation

- 8.380.2.1 `uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pCodecSTGain`
- 8.380.2.2 `uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pDTMFTXGain`
- 8.380.2.3 `uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pECMode`
- 8.380.2.4 `uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pNSEnable`
- 8.380.2.5 `uint8_t pack_audio_SLQSSetAudioPathConfig_t::Profile`
- 8.380.2.6 `audio_RXAGCList* pack_audio_SLQSSetAudioPathConfig_t::pRXAGCList`
- 8.380.2.7 `uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pRXAVCAGCSwitch`
- 8.380.2.8 `audio_RXAVCList* pack_audio_SLQSSetAudioPathConfig_t::pRXAVCList`
- 8.380.2.9 `audio_RXPCMIIRFitr* pack_audio_SLQSSetAudioPathConfig_t::pRXPCMIIRFitr`
- 8.380.2.10 `audio_TXAGCList* pack_audio_SLQSSetAudioPathConfig_t::pTXAGCList`
- 8.380.2.11 `uint8_t* pack_audio_SLQSSetAudioPathConfig_t::pTXAVCSwitch`
- 8.380.2.12 `uint16_t* pack_audio_SLQSSetAudioPathConfig_t::pTXGain`
- 8.380.2.13 `audio_TXPCMIIRFitr* pack_audio_SLQSSetAudioPathConfig_t::pTXPCMIIRFitr`

## 8.381 `pack_audio_SLQSSetAudioProfile_t` Struct Reference

### Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [EarMute](#)
- `uint8_t` [MicMute](#)
- `uint8_t` [Generator](#)
- `uint8_t` [Volume](#)

### 8.381.1 Detailed Description

This structure contains pack parameters to Set Audio Profile.

#### 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>

### 8.381.2 Field Documentation

8.381.2.1 uint8\_t pack\_audio\_SLQSSetAudioProfile\_t::EarMute

8.381.2.2 uint8\_t pack\_audio\_SLQSSetAudioProfile\_t::Generator

8.381.2.3 uint8\_t pack\_audio\_SLQSSetAudioProfile\_t::MicMute

8.381.2.4 uint8\_t pack\_audio\_SLQSSetAudioProfile\_t::Profile

8.381.2.5 uint8\_t pack\_audio\_SLQSSetAudioProfile\_t::Volume

## 8.382 pack\_audio\_SLQSSetAudioVoITLBConfig\_t Struct Reference

### Data Fields

- uint8\_t [Profile](#)
- uint8\_t [Generator](#)
- uint8\_t [Volume](#)
- uint8\_t [Item](#)
- uint16\_t [VolValue](#)

### 8.382.1 Detailed Description

This structure contains the pack parameters for SLQSSetAudioVoITLBConfig.

#### 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>

### 8.382.2 Field Documentation

8.382.2.1 uint8\_t pack\_audio\_SLQSSetAudioVoITLBConfig\_t::Generator

8.382.2.2 uint8\_t pack\_audio\_SLQSSetAudioVoITLBConfig\_t::Item

8.382.2.3 uint8\_t pack\_audio\_SLQSSetAudioVoITLBConfig\_t::Profile

8.382.2.4 uint8\_t pack\_audio\_SLQSSetAudioVoITLBConfig\_t::Volume

8.382.2.5 uint16\_t pack\_audio\_SLQSSetAudioVoITLBConfig\_t::VolValue

## 8.383 pack\_cat\_CATSendEnvelopeCommand\_t Struct Reference

## Data Fields

- uint32\_t [cmdID](#)
- uint32\_t [dataLen](#)
- uint8\_t \* [pData](#)

### 8.383.1 Detailed Description

Structure used to store envelope command data.

#### 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</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>

### 8.383.2 Field Documentation

8.383.2.1 uint32\_t pack\_cat\_CATSendEnvelopeCommand\_t::cmdID

8.383.2.2 uint32\_t pack\_cat\_CATSendEnvelopeCommand\_t::dataLen

8.383.2.3 uint8\_t\* pack\_cat\_CATSendEnvelopeCommand\_t::pData

## 8.384 pack\_cat\_CATSendTerminalResponse\_t Struct Reference

## Data Fields

- uint32\_t [refID](#)
- uint32\_t [dataLen](#)
- uint8\_t \* [pData](#)

### 8.384.1 Detailed Description

Structure used to terminal response data.

#### 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</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>

### 8.384.2 Field Documentation

8.384.2.1 `uint32_t pack_cat_CATSendTerminalResponse_t::dataLen`

8.384.2.2 `uint8_t* pack_cat_CATSendTerminalResponse_t::pData`

8.384.2.3 `uint32_t pack_cat_CATSendTerminalResponse_t::refID`

## 8.385 `pack_cat_SetCATEventCallback_t` Struct Reference

### Data Fields

- `uint32_t eventMask`

### 8.385.1 Detailed Description

This structure contains sEnables/disables the CAT event callback pack variable.

#### Parameters

<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>
------------------	---

### 8.385.2 Field Documentation

8.385.2.1 `uint32_t pack_cat_SetCATEventCallback_t::eventMask`



## 8.386 pack\_dms\_ActivateAutomatic\_t Struct Reference

### Data Fields

- uint8\_t [actCode](#) [81]

### 8.386.1 Detailed Description

This structure contains UIM activate automation information

#### Parameters

<i>actCode</i>	<ul style="list-style-type: none"> <li>• NULL-terminated string representing activation code (maximum string length of 12).</li> <li>• Specific carrier requirements may dictate actual activation code that is applicable, e.g., "*22899"</li> </ul>
----------------	---

### 8.386.2 Field Documentation

8.386.2.1 uint8\_t pack\_dms\_ActivateAutomatic\_t::actCode[81]

## 8.387 pack\_dms\_GetCustFeaturesV2\_t Struct Reference

### Data Fields

- uint8\_t [cust\\_id](#) [64+1]
- uint8\_t [list\\_type](#)
- uint16\_t [Tlvresult](#)

### 8.387.1 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> <li>• NULL terminated ASCII string.</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>

### 8.387.2 Field Documentation

8.387.2.1 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::cust\_id[64+1]

8.387.2.2 uint8\_t pack\_dms\_GetCustFeaturesV2\_t::list\_type

8.387.2.3 uint16\_t pack\_dms\_GetCustFeaturesV2\_t::Tlvresult

## 8.388 pack\_dms\_ResetToFactoryDefaults\_t Struct Reference

### Data Fields

- uint8\_t [spc](#) [6]

### 8.388.1 Detailed Description

This structure contains UIM reset to factory default information

#### Parameters

<i>spc</i> [IN]	<ul style="list-style-type: none"> <li>• Service programming code in ASCII format (digits 0 to 9 only)</li> </ul>
-----------------	---

### 8.388.2 Field Documentation

8.388.2.1 uint8\_t pack\_dms\_ResetToFactoryDefaults\_t::spc[6]

## 8.389 pack\_dms\_SetActivationStatusCallback\_t Struct Reference

### Data Fields

- uint8\_t [activationState](#)

### 8.389.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter pack.

#### Parameters

<i>activationState</i>	<ul style="list-style-type: none"> <li>• Service activation state.</li> <li>• Values             <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report activation state changes</li> </ul> </li> </ul>
------------------------	---

### 8.389.2 Field Documentation

8.389.2.1 uint8\_t pack\_dms\_SetActivationStatusCallback\_t::activationState

## 8.390 pack\_dms\_SetCrashAction\_t Struct Reference

### Data Fields

- uint8\_t [crashAction](#)

### 8.390.1 Detailed Description

This structure is used to store pack\_dms\_SetCrashAction parameters.

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

### 8.390.2 Field Documentation

8.390.2.1 uint8\_t pack\_dms\_SetCrashAction\_t::crashAction

## 8.391 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](#)

### 8.391.1 Detailed Description

This structure contains current settings of custom features pack parameters

#### Parameters

<i>GpsEnable</i>	<ul style="list-style-type: none"> <li>• describes if GPS is enabled or disabled</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - GPS is disabled</li> <li>– 0x01 - GPS is enabled</li> </ul> </li> </ul>
<i>DisableIMSI</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> </ul>

<i>IPFamSupport</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> </ul>
<i>RMAutoConnect</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> </ul>
<i>GPSSel</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 Port</li> <li>– 0x01 - GPS Rx over AUX Port</li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> </ul>
<i>SMSSupport</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> </ul>
<i>IsVoiceEnabled</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>DHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCP Relay support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Disable DHCP relay</li> <li>– 0x01 - Enable DHCP relay</li> </ul> </li> </ul>

<i>GPSLPM</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>
---------------	---

## 8.391.2 Field Documentation

8.391.2.1 `uint8_t pack_dms_SetCustFeature_t::DHCPRelayEnabled`

8.391.2.2 `uint8_t pack_dms_SetCustFeature_t::DisableIMSI`

8.391.2.3 `uint32_t pack_dms_SetCustFeature_t::GpsEnable`

8.391.2.4 `uint8_t pack_dms_SetCustFeature_t::GPSLPM`

8.391.2.5 `uint8_t pack_dms_SetCustFeature_t::GPSSel`

8.391.2.6 `uint16_t pack_dms_SetCustFeature_t::IPFamSupport`

8.391.2.7 `uint8_t pack_dms_SetCustFeature_t::IsVoiceEnabled`

8.391.2.8 `uint8_t pack_dms_SetCustFeature_t::RMAutoConnect`

8.391.2.9 `uint8_t pack_dms_SetCustFeature_t::SMSSupport`

## 8.392 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`

### 8.392.1 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> <li>• NULL terminated ASCII string.</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>
------------------	---

### 8.392.2 Field Documentation

8.392.2.1 uint8\_t pack\_dms\_SetCustFeaturesV2\_t::cust\_id[64+1]

8.392.2.2 uint8\_t pack\_dms\_SetCustFeaturesV2\_t::cust\_value[8+1]

8.392.2.3 uint16\_t pack\_dms\_SetCustFeaturesV2\_t::Tlvresult

8.392.2.4 uint16\_t pack\_dms\_SetCustFeaturesV2\_t::value\_length

## 8.393 pack\_dms\_SetEventReport\_t Struct Reference

### Data Fields

- uint8\_t [mode](#)

### 8.393.1 Detailed Description

This structure is used to store pack\_dms\_SetEventReport parameters.

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>• Operating Mode</li> <li>• 0 - reset</li> <li>• 1 - set</li> <li>• 2 - no change</li> </ul>
-------------	---

### 8.393.2 Field Documentation

8.393.2.1 uint8\_t pack\_dms\_SetEventReport\_t::mode

## 8.394 pack\_dms\_SetIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t [PSMStatus](#)
- uint8\_t [PSMCfgChangeInfo](#)
- uint8\_t [RptIMSCapability](#)

### 8.394.1 Detailed Description

This structure contains the DMS Register Indication request parameters

## Parameters

<i>PSMStatus</i>	<ul style="list-style-type: none"> <li>• Power Save Mode Status</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report PSM status</li> <li>– 2 - No Change</li> </ul> </li> </ul>
<i>PSMCfgChange-Info</i>	<ul style="list-style-type: none"> <li>• Power Save Mode Configuration Change Information</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report Power Save Mode configuration changes</li> <li>– 2 - No Change</li> </ul> </li> </ul>
<i>RptIMS-Capability</i>	<ul style="list-style-type: none"> <li>• Report IMS Capability</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Do not report</li> <li>– 1 - Report IMS capability</li> <li>– 2 - No Change</li> </ul> </li> </ul>

## 8.394.2 Field Documentation

8.394.2.1 uint8\_t pack\_dms\_SetIndicationRegister\_t::PSMCfgChangeInfo

8.394.2.2 uint8\_t pack\_dms\_SetIndicationRegister\_t::PSMStatus

8.394.2.3 uint8\_t pack\_dms\_SetIndicationRegister\_t::RptIMSCapability

## 8.395 pack\_dms\_SetPower\_t Struct Reference

## Data Fields

- uint32\_t [mode](#)
- uint16\_t [Tlvresult](#)

## 8.395.1 Detailed Description

Structure to store pack the operating mode of the device.

## Parameters

<i>mode</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>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• pack Tlv Result.</li> </ul>

## 8.395.2 Field Documentation

8.395.2.1 uint32\_t pack\_dms\_SetPower\_t::mode

8.395.2.2 uint16\_t pack\_dms\_SetPower\_t::Tlvresult

## 8.396 pack\_dms\_SetUSBComp\_t Struct Reference

### Data Fields

- uint8\_t [USBComp](#)
- uint16\_t [Tlvresult](#)

### 8.396.1 Detailed Description

This structure is used to store USB composition information pack paramters

#### Parameters

<i>USBComp</i>	<ul style="list-style-type: none"> <li>• Current USB Composition</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0..5 - Reserved (non-QMI)</li> <li>– 6 - DM NMEA AT QMI</li> <li>– 7 - DM NMEA AT QMI1 QMI2 QMI3</li> <li>– 8 - DM NMEA AT MBIM</li> <li>– 9 - MBIM</li> <li>– 10 - NMEA MBIM</li> <li>– 11 - DM MBIM</li> <li>– 12 - DM NMEA MBIM</li> <li>13-22 are combined compositions. One is for Win8 MBIM interfaces, another is for legacy QMI interfaces</li> <li>– 13 - 6 for QMI, 8 for MBIM</li> <li>– 14 - 6 for QMI, 9 for MBIM</li> <li>– 15 - 6 for QMI, 10 for MBIM</li> <li>– 16 - 6 for QMI, 11 for MBIM</li> <li>– 17 - 6 for QMI, 12 for MBIM</li> <li>– 18 - 7 for QMI, 8 for MBIM</li> <li>– 19 - 7 for QMI, 9 for MBIM</li> <li>– 20 - 7 for QMI, 10 for MBIM</li> <li>– 21 - 7 for QMI, 11 for MBIM</li> <li>– 22 - 7 for QMI, 12 for MBIM</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• pack Tlv Result.</li> </ul>

## 8.396.2 Field Documentation

8.396.2.1 uint16\_t pack\_dms\_SetUSBComp\_t::Tlvresult



8.396.2.2 uint8\_t pack\_dms\_SetUSBComp\_t::USBComp

## 8.397 pack\_dms\_SLQSDmsSwilIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t [resetInfoInd](#)

### 8.397.1 Detailed Description

This structure contains set registration state for different indication pack

#### Parameters

<i>resetInfoInd[IN]</i>	<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>
-------------------------	---

### 8.397.2 Field Documentation

8.397.2.1 uint8\_t pack\_dms\_SLQSDmsSwilIndicationRegister\_t::resetInfoInd

## 8.398 pack\_dms\_SLQSSetPowerSaveModeConfig\_t Struct Reference

### Data Fields

- uint8\_t \* [pPsmEnableState](#)
- uint32\_t \* [pActiveTimer](#)
- uint32\_t \* [pPeriodicUpdateTimer](#)

### 8.398.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

#### Parameters

<i>pPsmEnable-State</i>	[Optional] <ul style="list-style-type: none"> <li>PSM Enable State</li> <li>Values               <ul style="list-style-type: none"> <li>0 - PSM is not enabled</li> <li>1 - PSM is enabled</li> </ul> </li> </ul>
<i>pActiveTimer</i>	[Optional] <ul style="list-style-type: none"> <li>Active Timer</li> <li>PSM active timer value (in seconds).</li> <li>ActiveTimer encoding is the GPRS timer 2 encoding specified in Table 10.5.172 of 3GPP TS 24.008.</li> </ul>

<i>pPeriodic-UpdateTimer</i>	[Optional] <ul style="list-style-type: none"> <li>• Periodic Update Timer</li> <li>• PSM periodic update timer value (in seconds)</li> <li>• Periodic Update Timer encoding is the GPRS timer 3 encoding specified in Table 10.5.163a of 3GPP TS 24.008.</li> </ul>
------------------------------	---

## 8.398.2 Field Documentation

8.398.2.1 uint32\_t\* pack\_dms\_SLQSSetPowerSaveModeConfig\_t::pActiveTimer

8.398.2.2 uint32\_t\* pack\_dms\_SLQSSetPowerSaveModeConfig\_t::pPeriodicUpdateTimer

8.398.2.3 uint8\_t\* pack\_dms\_SLQSSetPowerSaveModeConfig\_t::pPsmEnableState

## 8.399 pack\_dms\_SLQSSwiGetCrashInfo\_t Struct Reference

### Data Fields

- uint8\_t [clear](#)

### 8.399.1 Detailed Description

This structure contains pack\_dms\_SLQSSwiGetCrashInfo parameters

#### Parameters

<i>clear</i>	<ul style="list-style-type: none"> <li>• 0 - do not clear crash data after response</li> <li>• 1 - clear crash data after response</li> </ul>
--------------	---

## 8.399.2 Field Documentation

8.399.2.1 uint8\_t pack\_dms\_SLQSSwiGetCrashInfo\_t::clear

## 8.400 pack\_dms\_SLQSSwiSetDyingGaspCfg\_t Struct Reference

### Data Fields

- uint8\_t \* [pDestSMSNum](#)
- uint8\_t \* [pDestSMSContent](#)

### 8.400.1 Detailed Description

This structure is used to store pack\_dms\_SLQSSwiSetDyingGaspCfg parameters

## Parameters

<i>pDestSMSNum</i> [ <i>IN</i> ]	<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> [ <i>IN</i> ]	<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>

## 8.400.2 Field Documentation

8.400.2.1 uint8\_t\* pack\_dms\_SLQSSwiSetDyingGaspCfg\_t::pDestSMSContent

8.400.2.2 uint8\_t\* pack\_dms\_SLQSSwiSetDyingGaspCfg\_t::pDestSMSNum

## 8.401 pack\_dms\_SLQSSwiSetHostDevInfo\_t Struct Reference

## Data Fields

- char [manString](#) [255]
- char [modelString](#) [255]
- char [swVerString](#) [255]
- char [plasmaIDString](#) [255]
- char [hostID](#) [255]

## 8.401.1 Detailed Description

This structure contains SWI set host device info unpack information

## Parameters

<i>manString</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>Host Device Manufacturer Name</li> <li>Null terminated ASCII String</li> </ul>
<i>modelString</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>Host Device Model String</li> <li>Null terminated ASCII string.</li> </ul>
<i>swVerString</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>Host Device Software Version String</li> <li>Null terminated ASCII string</li> </ul>
<i>plasmaIDString</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>Host Device Plasma ID String</li> <li>Null terminated alphanumeric ASCII string</li> </ul>
<i>HostID</i> [ <i>IN</i> ]	<ul style="list-style-type: none"> <li>Device Host ID String</li> <li>Null terminated alphanumeric ASCII string</li> </ul>

### 8.401.2 Field Documentation

8.401.2.1 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::hostID[255]

8.401.2.2 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::manString[255]

8.401.2.3 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::modelString[255]

8.401.2.4 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::plasmaIDString[255]

8.401.2.5 char pack\_dms\_SLQSSwiSetHostDevInfo\_t::swVerString[255]

## 8.402 pack\_dms\_SLQSSwiSetOSInfo\_t Struct Reference

### Data Fields

- char [nameString](#) [255]
- char [versionString](#) [255]

### 8.402.1 Detailed Description

This structure contains SWI set host OS info pack information

#### Parameters

<i>nameString</i> [IN]	Host device manufacture String <ul style="list-style-type: none"> <li>• NULL terminated ASCII string.</li> </ul>
<i>VersionString</i> [IN]	Host device model String <ul style="list-style-type: none"> <li>• NULL terminated ASCII string.</li> </ul>

### 8.402.2 Field Documentation

8.402.2.1 char pack\_dms\_SLQSSwiSetOSInfo\_t::nameString[255]

8.402.2.2 char pack\_dms\_SLQSSwiSetOSInfo\_t::versionString[255]

## 8.403 pack\_dms\_SwiSetEventReport\_t Struct Reference

### Data Fields

- uint8\_t \* [pTempReport](#)
- uint8\_t \* [pVoltReport](#)
- uint8\_t \* [pUIMStatusReport](#)

### 8.403.1 Detailed Description

This structure is used to store pack\_dms\_SwiSetEventReport parameters.

## Parameters

<i>pTempReport</i>	(optional) <ul style="list-style-type: none"> <li>Temperature indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report when state changes</li> </ul> </li> </ul>
<i>pVoltReport</i>	(optional) <ul style="list-style-type: none"> <li>Voltage Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report when state changes</li> </ul> </li> </ul>
<i>pUIMStatus-Report</i>	(optional) <ul style="list-style-type: none"> <li>UIM status Indicator. <ul style="list-style-type: none"> <li>0 - Do not report</li> <li>1 - Report when state changes</li> </ul> </li> </ul>

## 8.403.2 Field Documentation

8.403.2.1 uint8\_t\* pack\_dms\_SwiSetEventReport\_t::pTempReport

8.403.2.2 uint8\_t\* pack\_dms\_SwiSetEventReport\_t::pUIMStatusReport

8.403.2.3 uint8\_t\* pack\_dms\_SwiSetEventReport\_t::pVoltReport

## 8.404 pack\_dms\_SwiUimSelect\_t Struct Reference

## Data Fields

- uint8\_t [uim\\_select](#)

## 8.404.1 Detailed Description

This structure is used to store pack\_dms\_SwiUimSelect parameters.

## Parameters

<i>uim_select</i>	<ul style="list-style-type: none"> <li>Selection of UIM Card <ul style="list-style-type: none"> <li>0 - External UIM Interface</li> <li>1 - Embedded UIM</li> <li>2 - Remote UIM</li> <li>3 - Auto switch</li> </ul> </li> </ul>
-------------------	--

## Note

Auto switch is applicable only if UIMAUTOSWITCH feature is enabled. This feature will activate the auto-switch mechanism if it is inactive.

## 8.404.2 Field Documentation

8.404.2.1 `uint8_t pack_dms_SwiUimSelect_t::uim_select`

## 8.405 `pack_dms_UIMChangePIN_t` Struct Reference

### Data Fields

- `uint8_t id`
- `uint8_t oldValue` [255]
- `uint8_t newValue` [255]

### 8.405.1 Detailed Description

This structure contains UIM Unblock PIN Information

#### 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>oldValue</i> [IN]	<ul style="list-style-type: none"> <li>• Old PIN value of PIN to change</li> </ul>
<i>newValue</i> [IN]	<ul style="list-style-type: none"> <li>• New PIN value of PIN to change</li> </ul>

### 8.405.2 Field Documentation

8.405.2.1 `uint8_t pack_dms_UIMChangePIN_t::id`

8.405.2.2 `uint8_t pack_dms_UIMChangePIN_t::newValue`[255]

8.405.2.3 `uint8_t pack_dms_UIMChangePIN_t::oldValue`[255]

## 8.406 `pack_dms_UIMGetControlKeyStatus_t` Struct Reference

### Data Fields

- `uint8_t facility`

### 8.406.1 Detailed Description

This structure contains UIM get control key status information

## Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> <li>• MT or network facility <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>
----------------------	--

## 8.406.2 Field Documentation

8.406.2.1 uint8\_t pack\_dms\_UIMGetControlKeyStatus\_t::facility

## 8.407 pack\_dms\_UIMGetICCID\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.407.1 Detailed Description

This structure contains UIM Get ICCID pack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Pack result.</li> </ul>
------------------	--

## 8.407.2 Field Documentation

8.407.2.1 [swi\\_uint256\\_t](#) pack\_dms\_UIMGetICCID\_t::ParamPresenceMask

8.407.2.2 uint16\_t pack\_dms\_UIMGetICCID\_t::Tlvresult

## 8.408 pack\_dms\_UIMSetControlKeyProtection\_t Struct Reference

## Data Fields

- uint8\_t [facility](#)
- uint8\_t [facilityState](#)
- uint8\_t [facilityCk](#) [8]

## 8.408.1 Detailed Description

This structure contains UIM Set control key protection information

## Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> <li>• MT or network facility <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>facilityState</i> [IN]	<ul style="list-style-type: none"> <li>• UIM facility state <ul style="list-style-type: none"> <li>– 0 - Deactivated</li> <li>– 1 - Activated</li> <li>– 2 - Blocked</li> </ul> </li> </ul>
<i>facilityCk</i> [IN]	<ul style="list-style-type: none"> <li>• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)</li> </ul>

## 8.408.2 Field Documentation

8.408.2.1 uint8\_t pack\_dms\_UIMSetControlKeyProtection\_t::facility

8.408.2.2 uint8\_t pack\_dms\_UIMSetControlKeyProtection\_t::facilityCk[8]

8.408.2.3 uint8\_t pack\_dms\_UIMSetControlKeyProtection\_t::facilityState

## 8.409 pack\_dms\_UIMSetPINProtection\_t Struct Reference

## Data Fields

- uint8\_t [id](#)
- uint8\_t [bEnable](#)
- uint8\_t [value](#) [255]

## 8.409.1 Detailed Description

This structure contains PIN Protection Information

## 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>



## 8.409.2 Field Documentation

8.409.2.1 uint8\_t pack\_dms\_UIMSetPINProtection\_t::bEnable

8.409.2.2 uint8\_t pack\_dms\_UIMSetPINProtection\_t::id

8.409.2.3 uint8\_t pack\_dms\_UIMSetPINProtection\_t::value[255]

## 8.410 pack\_dms\_UIMUnlockControlKey\_t Struct Reference

### Data Fields

- uint8\_t [facility](#)
- uint8\_t [facilityCk](#) [8]

### 8.410.1 Detailed Description

This structure contains UIM unlock Control Key information

#### Parameters

<i>facility</i> [IN]	<ul style="list-style-type: none"> <li>• MT or network facility <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>facilityCk</i> [IN]	<ul style="list-style-type: none"> <li>• Facility depersonalization control key, string in ASCII text (maximum 8 bytes)</li> </ul>

## 8.410.2 Field Documentation

8.410.2.1 uint8\_t pack\_dms\_UIMUnlockControlKey\_t::facility

8.410.2.2 uint8\_t pack\_dms\_UIMUnlockControlKey\_t::facilityCk[8]

## 8.411 pack\_dms\_UIMUnlockPIN\_t Struct Reference

### Data Fields

- uint8\_t [id](#)
- uint8\_t [pukValue](#) [255]
- uint8\_t [newPin](#) [255]

### 8.411.1 Detailed Description

This structure contains UIM Unblock PIN Information

## 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>pukValue</i> [IN]	<ul style="list-style-type: none"> <li>PUK value of PIN to be unblocked</li> </ul>
<i>newPin</i> [IN]	<ul style="list-style-type: none"> <li>New PIN value for the PIN to be unblocked</li> </ul>

## 8.411.2 Field Documentation

8.411.2.1 uint8\_t pack\_dms\_UIMUnblockPIN\_t::id

8.411.2.2 uint8\_t pack\_dms\_UIMUnblockPIN\_t::newPin[255]

8.411.2.3 uint8\_t pack\_dms\_UIMUnblockPIN\_t::pukValue[255]

## 8.412 pack\_dms\_UIMVerifyPIN\_t Struct Reference

## Data Fields

- uint8\_t [id](#)
- uint8\_t [value](#) [255]

## 8.412.1 Detailed Description

This structure contains PIN Value Information

## 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>value</i> [IN]	<ul style="list-style-type: none"> <li>PIN value of the PIN to be enabled/disabled</li> </ul>

## 8.412.2 Field Documentation

8.412.2.1 uint8\_t pack\_dms\_UIMVerifyPIN\_t::id

8.412.2.2 uint8\_t pack\_dms\_UIMVerifyPIN\_t::value[255]

## 8.413 pack\_fms\_GetImagesPreference\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.413.1 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>
------------------	---

### 8.413.2 Field Documentation

8.413.2.1 uint16\_t pack\_fms\_GetImagesPreference\_t::Tlvresult

## 8.414 pack\_fms\_GetStoredImages\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)

### 8.414.1 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>
------------------	---

### 8.414.2 Field Documentation

8.414.2.1 uint16\_t pack\_fms\_GetStoredImages\_t::Tlvresult

## 8.415 pack\_fms\_SetImagesPreference\_t Struct Reference

## Data Fields

- uint32\_t [imageListSize](#)
- [FMSPrefImageList](#) \* [pImageList](#)
- uint32\_t [bForceDownload](#)
- uint8\_t [modemindex](#)
- uint16\_t [Tlvresult](#)

### 8.415.1 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>plmageList</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 - Focrce 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>

## 8.415.2 Field Documentation

8.415.2.1 uint32\_t pack\_fms\_SetImagesPreference\_t::bForceDownload

8.415.2.2 uint32\_t pack\_fms\_SetImagesPreference\_t::imageListSize

8.415.2.3 uint8\_t pack\_fms\_SetImagesPreference\_t::modemindex

8.415.2.4 FMSPrefImageList\* pack\_fms\_SetImagesPreference\_t::plmageList

8.415.2.5 uint16\_t pack\_fms\_SetImagesPreference\_t::Tlvresult

## 8.416 pack\_ims\_SLQSImsConfigIndicationRegister\_t Struct Reference

## Data Fields

- uint8\_t \* [pSIPConfigEvents](#)
- uint8\_t \* [pRegMgrConfigEvents](#)
- uint8\_t \* [pSMSCConfigEvents](#)
- uint8\_t \* [pUserConfigEvents](#)
- uint8\_t \* [pVoIPConfigEvents](#)

## 8.416.1 Detailed Description

This structure contains parameters of IMS Config Indication Register

## Parameters

<i>pSIPConfigEvents(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.

**8.416.2 Field Documentation**

8.416.2.1 uint8\_t\* pack\_ims\_SLQSImsConfigIndicationRegister\_t::pRegMgrConfigEvents

8.416.2.2 uint8\_t\* pack\_ims\_SLQSImsConfigIndicationRegister\_t::pSIPConfigEvents

8.416.2.3 uint8\_t\* pack\_ims\_SLQSImsConfigIndicationRegister\_t::pSMSConfigEvents

8.416.2.4 uint8\_t\* pack\_ims\_SLQSImsConfigIndicationRegister\_t::pUserConfigEvents

8.416.2.5 uint8\_t\* pack\_ims\_SLQSImsConfigIndicationRegister\_t::pVoIPConfigEvents

**8.417 pack\_ims\_SLQSSetIMSSMSConfig\_t Struct Reference****Data Fields**

- uint8\_t \* [pSMSFormat](#)
- uint8\_t \* [pSMSOverlPNwInd](#)
- uint8\_t \* [pPhoneCtxURLen](#)
- uint8\_t \* [pPhoneCtxURI](#)

### 8.417.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig pack 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>

### 8.417.2 Field Documentation

8.417.2.1 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURI`

8.417.2.2 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pPhoneCtxtURLen`

8.417.2.3 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSFormat`

8.417.2.4 `uint8_t* pack_ims_SLQSSetIMSSMSConfig_t::pSMSOverIPNwInd`

## 8.418 pack\_ims\_SLQSSetIMSUserConfig\_t Struct Reference

### Data Fields

- `uint8_t * pIMSDomainLen`
- `uint8_t * pIMSDomain`

### 8.418.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig pack parameters.

#### Parameters

<i>pIMSDomainLen</i>	<ul style="list-style-type: none"> <li>Length of IMS Domain Name to follow</li> </ul>
<i>pIMSDomain</i>	<ul style="list-style-type: none"> <li>IMS domain name</li> </ul>

## 8.418.2 Field Documentation

8.418.2.1 uint8\_t\* pack\_ims\_SLQSSetIMSUserConfig\_t::pIMSDomain

8.418.2.2 uint8\_t\* pack\_ims\_SLQSSetIMSUserConfig\_t::pIMSDomainLen

## 8.419 pack\_ims\_SLQSSetIMSVoIPConfig\_t Struct Reference

### Data Fields

- uint16\_t \* pSessionExpiryTimer
- uint16\_t \* pMinSessionExpiryTimer
- uint8\_t \* pAmrWbEnable
- uint8\_t \* pScrAmrEnable
- uint8\_t \* pScrAmrWbEnable
- uint8\_t \* pAmrMode
- uint16\_t \* pAmrWBMode
- uint8\_t \* pAmrOctetAligned
- uint8\_t \* pAmrWBOctetAligned
- uint16\_t \* pRingingTimer
- uint16\_t \* pRingBackTimer
- uint16\_t \* pRTPRTCPInactTimer

### 8.419.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig pack 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>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 - 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>



## 8.419.2 Field Documentation

- 8.419.2.1 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pAmrMode
- 8.419.2.2 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pAmrOctetAligned
- 8.419.2.3 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pAmrWbEnable
- 8.419.2.4 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pAmrWBMode
- 8.419.2.5 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pAmrWBOctetAligned
- 8.419.2.6 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pMinSessionExpiryTimer
- 8.419.2.7 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pRingBackTimer
- 8.419.2.8 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pRingingTimer
- 8.419.2.9 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pRTPRTCPlnactTimer
- 8.419.2.10 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pScrAmrEnable
- 8.419.2.11 uint8\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pScrAmrWbEnable
- 8.419.2.12 uint16\_t\* pack\_ims\_SLQSSetIMSVolIPConfig\_t::pSessionExpiryTimer

## 8.420 pack\_ims\_SLQSSetRegMgrConfig\_t Struct Reference

### Data Fields

- uint16\_t \* pPriCSCFPort
- uint8\_t \* pCSCFPortNameLen
- uint8\_t \* pCSCFPortName
- uint8\_t \* pIMSTestMode

### 8.420.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig pack parameters.

#### Parameters

<i>pPriCSCFPort</i>	<ul style="list-style-type: none"> <li>• Primary call session control function port</li> </ul>
<i>pCSCFPort-NameLen</i>	<ul style="list-style-type: none"> <li>• Length of the CSCF 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>

## 8.420.2 Field Documentation

8.420.2.1 uint8\_t\* pack\_ims\_SLQSSetRegMgrConfig\_t::pCSCFPortName

8.420.2.2 uint8\_t\* pack\_ims\_SLQSSetRegMgrConfig\_t::pCSCFPortNameLen

8.420.2.3 uint8\_t\* pack\_ims\_SLQSSetRegMgrConfig\_t::pIMSTestMode

8.420.2.4 uint16\_t\* pack\_ims\_SLQSSetRegMgrConfig\_t::pPriCSCFPort

## 8.421 pack\_ims\_SLQSSetSIPConfig\_t Struct Reference

### Data Fields

- uint16\_t \* [pSIPLocalPort](#)
- uint32\_t \* [pTimerSIPReg](#)
- uint32\_t \* [pSubscribeTimer](#)
- uint32\_t \* [pTimerT1](#)
- uint32\_t \* [pTimerT2](#)
- uint32\_t \* [pTimerTf](#)
- uint8\_t \* [pSigCompEnabled](#)

### 8.421.1 Detailed Description

This structure contains the SLQSSetSIPConfig pack 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>

### 8.421.2 Field Documentation

8.421.2.1 uint8\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pSigCompEnabled

8.421.2.2 uint16\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pSIPLocalPort

8.421.2.3 uint32\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pSubscribeTimer

8.421.2.4 uint32\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pTimerSIPReg

8.421.2.5 uint32\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pTimerT1

8.421.2.6 uint32\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pTimerT2

8.421.2.7 uint32\_t\* pack\_ims\_SLQSSetSIPConfig\_t::pTimerTf

## 8.422 pack\_imsa\_SLQSRegisterIMSAIndication\_t Struct Reference

### Data Fields

- int [has\\_RegStatusConfig](#)
- uint8\_t [RegStatusConfig](#)
- int [has\\_ServiceStatusConfig](#)
- uint8\_t [ServiceStatusConfig](#)
- int [has\\_RatHandoverStatusConfig](#)
- uint8\_t [RatHandoverStatusConfig](#)
- int [has\\_PdpStatusConfig](#)
- uint8\_t [PdpStatusConfig](#)

### 8.422.1 Detailed Description

This structure contains parameters of IMSA Config Indication Register

#### Parameters

<i>RegStatus-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>ServiceStatus-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>RatHandover-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>PdpStatus-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.

**8.422.2 Field Documentation**

8.422.2.1 int pack\_imsa\_SLQSRegisterIMSAIndication\_t::has\_PdpStatusConfig

8.422.2.2 int pack\_imsa\_SLQSRegisterIMSAIndication\_t::has\_RatHandoverStatusConfig

8.422.2.3 int pack\_imsa\_SLQSRegisterIMSAIndication\_t::has\_RegStatusConfig

8.422.2.4 int pack\_imsa\_SLQSRegisterIMSAIndication\_t::has\_ServiceStatusConfig

8.422.2.5 uint8\_t pack\_imsa\_SLQSRegisterIMSAIndication\_t::PdpStatusConfig

8.422.2.6 uint8\_t pack\_imsa\_SLQSRegisterIMSAIndication\_t::RatHandoverStatusConfig

8.422.2.7 uint8\_t pack\_imsa\_SLQSRegisterIMSAIndication\_t::RegStatusConfig

8.422.2.8 uint8\_t pack\_imsa\_SLQSRegisterIMSAIndication\_t::ServiceStatusConfig

**8.423 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](#)

**8.423.1 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>

## 8.423.2 Field Documentation

8.423.2.1 `loc_BdsSVInfo* pack_loc_Delete_Assist_Data_t::pBdsSVInfo`

8.423.2.2 `loc_CellDb* pack_loc_Delete_Assist_Data_t::pCellDb`

8.423.2.3 `loc_ClkInfo* pack_loc_Delete_Assist_Data_t::pClkInfo`

8.423.2.4 `loc_GnssData* pack_loc_Delete_Assist_Data_t::pGnssData`

8.423.2.5 `loc_SVInfo* pack_loc_Delete_Assist_Data_t::pSVInfo`

8.423.2.6 `uint16_t pack_loc_Delete_Assist_Data_t::Tlvresult`

## 8.424 pack\_loc\_EventRegister\_t Struct Reference

### Data Fields

- `uint64_t` [eventRegister](#)
- `uint16_t` [Tlvresult](#)

### 8.424.1 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, SV 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>
------------------	--

## 8.424.2 Field Documentation

8.424.2.1 uint64\_t pack\_loc\_EventRegister\_t::eventRegister

8.424.2.2 uint16\_t pack\_loc\_EventRegister\_t::Tlvresult

## 8.425 pack\_loc\_SetExtPowerState\_t Struct Reference

## Data Fields

- uint32\_t [extPowerState](#)
- uint16\_t [Tlvresult](#)

## 8.425.1 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>

## 8.425.2 Field Documentation

8.425.2.1 uint32\_t pack\_loc\_SetExtPowerState\_t::extPowerState

8.425.2.2 uint16\_t pack\_loc\_SetExtPowerState\_t::Tlvresult

## 8.426 pack\_loc\_SetOperationMode\_t Struct Reference

## Data Fields

- uint32\_t [mode](#)
- uint16\_t [Tlvresult](#)

### 8.426.1 Detailed Description

This structure contains Set Operation Mode pack

#### Parameters

<i>mode</i>	<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>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Pack result.</li> </ul>

### 8.426.2 Field Documentation

8.426.2.1 uint32\_t pack\_loc\_SetOperationMode\_t::mode

8.426.2.2 uint16\_t pack\_loc\_SetOperationMode\_t::Tlvresult

## 8.427 pack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

#### Data Fields

- uint32\_t [xid](#)
- uint16\_t [Tlvresult](#)

### 8.427.1 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>

### 8.427.2 Field Documentation

8.427.2.1 uint16\_t pack\_loc\_SLQSLOCGetBestAvailPos\_t::Tlvresult



8.427.2.2 uint32\_t pack\_loc\_SLQSLOCGetBestAvailPos\_t::xid

## 8.428 pack\_loc\_SLQSLOCGetServer\_t Struct Reference

### Data Fields

- uint32\_t [serverType](#)
- int [has\\_serverAddrTypeMask](#)
- uint8\_t [serverAddrTypeMask](#)

### 8.428.1 Detailed Description

This structure contains parameters to Gets the location server. Please check has\_<Param\_Name> field for presence of optional parameters

#### Parameters

<i>serverType</i>	<ul style="list-style-type: none"> <li>• Type of server.</li> <li>• Valid values <ul style="list-style-type: none"> <li>– eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE</li> <li>– eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC</li> <li>– eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP</li> <li>– eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4)- Server type is custom PDE</li> </ul> </li> </ul>
<i>serverAddrTypeMask</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Type of address the client wants. If unspecified, the indication will contain all the types of addresses it has for the specified server type.</li> <li>• Valid bitmasks <ul style="list-style-type: none"> <li>– 0x01 - IPv4</li> <li>– 0x02 - IPv6</li> <li>– 0x04 - URL</li> </ul> </li> </ul>

### 8.428.2 Field Documentation

8.428.2.1 int pack\_loc\_SLQSLOCGetServer\_t::has\_serverAddrTypeMask

8.428.2.2 uint8\_t pack\_loc\_SLQSLOCGetServer\_t::serverAddrTypeMask

8.428.2.3 uint32\_t pack\_loc\_SLQSLOCGetServer\_t::serverType

## 8.429 pack\_loc\_SLQSLOCInjectPosition\_t Struct Reference

### Data Fields

- double [latitude](#)
- int [has\\_latitude](#)
- double [longitude](#)
- int [has\\_longitude](#)
- float [horUncCircular](#)

- int [has\\_horUncCircular](#)
- uint8\_t [horConfidence](#)
- int [has\\_horConfidence](#)
- uint32\_t [horReliability](#)
- int [has\\_horReliability](#)
- float [altitudeWrtEllipsoid](#)
- int [has\\_altitudeWrtEllipsoid](#)
- float [altitudeWrtMeanSeaLevel](#)
- int [has\\_altitudeWrtMeanSeaLevel](#)
- float [vertUnc](#)
- int [has\\_vertUnc](#)
- uint8\_t [vertConfidence](#)
- int [has\\_vertConfidence](#)
- uint32\_t [vertReliability](#)
- int [has\\_vertReliability](#)
- altSrcInfo\_t [altitudeSrcInfo](#)
- int [has\\_altitudeSrcInfo](#)
- uint64\_t [timestampUtc](#)
- int [has\\_timestampUtc](#)
- uint32\_t [timestampAge](#)
- int [has\\_timestampAge](#)
- uint32\_t [positionSrc](#)
- int [has\\_positionSrc](#)
- float [rawHorUncCircular](#)
- int [has\\_rawHorUncCircular](#)
- uint8\_t [rawHorConfidence](#)
- int [has\\_rawHorConfidence](#)

### 8.429.1 Detailed Description

This structure contains LOC Inject Position parameters Please check has\_<Param\_Name> field for presence of optional parameters

#### Parameters

<i>latitude</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>longitude</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>horUncCircular</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>horConfidence</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>horReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter <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> </li> </ul>
<i>altitudeWrt-Ellipsoid</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>altitudeWrt-MeanSeaLevel</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>vertUnc</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 pAltitudeWrtEllipsoid or pAltitudeWrtMeanSeaLevel is specified.</li> </ul>

<i>vertConfidence</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>vertReliability</i>	<ul style="list-style-type: none"> <li>• Optional parameter <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> </li> </ul>
<i>altitudeSrcInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter <ul style="list-style-type: none"> <li>– Pointer to struct altitudeSrcInfo. See <a href="#">altitudeSrcInfo</a> for more information</li> </ul> </li> </ul>
<i>timestampUtc</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>timestampAge</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>positionSrc</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>rawHorUnc-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>rawHor-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>

## 8.429.2 Field Documentation

8.429.2.1 altSrcInfo\_t pack\_loc\_SLQSLOCInjectPosition\_t::altitudeSrcInfo

8.429.2.2 float pack\_loc\_SLQSLOCInjectPosition\_t::altitudeWrtEllipsoid

8.429.2.3 float pack\_loc\_SLQSLOCInjectPosition\_t::altitudeWrtMeanSeaLevel

8.429.2.4 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeSrcInfo

8.429.2.5 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeWrtEllipsoid

8.429.2.6 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_altitudeWrtMeanSeaLevel

8.429.2.7 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horConfidence

8.429.2.8 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horReliability

8.429.2.9 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_horUncCircular

8.429.2.10 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_latitude

8.429.2.11 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_longitude

8.429.2.12 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_positionSrc

8.429.2.13 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_rawHorConfidence

8.429.2.14 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_rawHorUncCircular

8.429.2.15 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_timestampAge

8.429.2.16 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_timestampUtc

8.429.2.17 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertConfidence

8.429.2.18 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertRelicability

8.429.2.19 int pack\_loc\_SLQSLOCInjectPosition\_t::has\_vertUnc

8.429.2.20 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::horConfidence

8.429.2.21 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::horReliability

8.429.2.22 float pack\_loc\_SLQSLOCInjectPosition\_t::horUncCircular

8.429.2.23 double pack\_loc\_SLQSLOCInjectPosition\_t::latitude

8.429.2.24 double pack\_loc\_SLQSLOCInjectPosition\_t::longitude

8.429.2.25 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::positionSrc

8.429.2.26 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::rawHorConfidence

8.429.2.27 float pack\_loc\_SLQSLOCInjectPosition\_t::rawHorUncCircular

8.429.2.28 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::timestampAge

8.429.2.29 uint64\_t pack\_loc\_SLQSLOCInjectPosition\_t::timestampUtc

8.429.2.30 uint8\_t pack\_loc\_SLQSLOCInjectPosition\_t::vertConfidence

8.429.2.31 uint32\_t pack\_loc\_SLQSLOCInjectPosition\_t::vertReliability

8.429.2.32 float pack\_loc\_SLQSLOCInjectPosition\_t::vertUnc

## 8.430 pack\_loc\_SLQSLOCInjectSensorData\_t Struct Reference

### Data Fields

- int [has\\_opaqueld](#)
- uint32\_t [opaqueld](#)
- int [has\\_accleroData](#)
- [sensorData\\_t](#) [accleroData](#)
- int [has\\_gyroData](#)
- [sensorData\\_t](#) [gyroData](#)
- int [has\\_acceleroTimeSrc](#)
- uint32\_t [acceleroTimeSrc](#)
- int [has\\_gyroTimeSrc](#)
- uint32\_t [gyroTimeSrc](#)
- int [has\\_accelTemp](#)
- [tempData\\_t](#) [accelTemp](#)
- int [has\\_gyroTemp](#)
- [tempData\\_t](#) [gyroTemp](#)

### 8.430.1 Detailed Description

This structure contains parameters to inject sensor data into the GNSS location engine Please check has\_<Param-Name> field for presence of optional parameters

## Parameters

<i>opaqueId</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>acceleroData</i>	<ul style="list-style-type: none"> <li>• 3-Axis Accelerometer Data (Optional parameter)</li> <li>• Pointer to struct sensorData. See <a href="#">sensorData_t</a> for more information</li> </ul>
<i>gyroData</i>	<ul style="list-style-type: none"> <li>• 3-Axis Gyroscope Data (Optional parameter)</li> <li>• Pointer to struct sensorData. See <a href="#">sensorData_t</a> for more information</li> </ul>
<i>acceleroTimeSrc</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>gyroTimeSrc</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>accelTemp</i>	<ul style="list-style-type: none"> <li>• Accelerometer Temperature Data (Optional parameter)</li> <li>• Pointer to struct tempratureData. See <a href="#">tempData_t</a> for more information</li> </ul>
<i>gyroTemp</i>	<ul style="list-style-type: none"> <li>• Gyroscope Temperature Data (Optional parameter)</li> <li>• Pointer to struct tempratureData. See <a href="#">tempData_t</a> for more information</li> </ul>

## 8.430.2 Field Documentation

8.430.2.1 `sensorData_t pack_loc_SLQSLOCInjectSensorData_t::acceleroData`8.430.2.2 `uint32_t pack_loc_SLQSLOCInjectSensorData_t::acceleroTimeSrc`8.430.2.3 `tempData_t pack_loc_SLQSLOCInjectSensorData_t::accelTemp`8.430.2.4 `sensorData_t pack_loc_SLQSLOCInjectSensorData_t::gyroData`

8.430.2.5 `tempData_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTemp`

8.430.2.6 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::gyroTimeSrc`

8.430.2.7 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_acceleroTimeSrc`

8.430.2.8 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accelTemp`

8.430.2.9 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_accleroData`

8.430.2.10 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroData`

8.430.2.11 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTemp`

8.430.2.12 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_gyroTimeSrc`

8.430.2.13 `int` `pack_loc_SLQSLOCInjectSensorData_t::has_opaqueld`

8.430.2.14 `uint32_t` `pack_loc_SLQSLOCInjectSensorData_t::opaqueld`

## 8.431 `pack_loc_SLQSLOCInjectUTCTime_t` Struct Reference

### Data Fields

- `uint64_t` [timeMsec](#)
- `uint32_t` [timeUncMsec](#)

### 8.431.1 Detailed Description

This structure contains inject UTC time parameter.

#### Parameters

<i>timeMsec</i>	<ul style="list-style-type: none"> <li>• The UTC time since Jan. 1, 1970</li> </ul>
<i>timeUncMsec</i>	<ul style="list-style-type: none"> <li>• The time Uncertainty</li> </ul>

### 8.431.2 Field Documentation

8.431.2.1 `uint64_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeMsec`

8.431.2.2 `uint32_t` `pack_loc_SLQSLOCInjectUTCTime_t::timeUncMsec`

## 8.432 `pack_loc_SLQSLOCSetCradleMountConfig_t` Struct Reference

### Data Fields

- `uint32_t` [state](#)
- `int` [has\\_confidence](#)
- `uint8_t` [confidence](#)



### 8.432.1 Detailed Description

This structure contains parameters to set current cradle mount configuration Please check has\_<Param\_Name> field for presence of optional parameters

#### 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>confidence</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>

### 8.432.2 Field Documentation

8.432.2.1 uint8\_t pack\_loc\_SLQSLOCSetCradleMountConfig\_t::confidence

8.432.2.2 int pack\_loc\_SLQSLOCSetCradleMountConfig\_t::has\_confidence

8.432.2.3 uint32\_t pack\_loc\_SLQSLOCSetCradleMountConfig\_t::state

## 8.433 pack\_loc\_SLQSLOCSetServer\_t Struct Reference

#### Data Fields

- uint32\_t [serverType](#)
- [loc\\_IPv4Config](#) \* [pIPv4Config](#)
- [loc\\_IPv6Config](#) \* [pIPv6Config](#)
- [loc\\_URLAddrInfo](#) \* [pURLAddr](#)

### 8.433.1 Detailed Description

This structure contains parameters to set A-GPS Server

#### Parameters

<i>serverType</i>	<ul style="list-style-type: none"> <li>• Type of server</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 1 - Server type is CDMA PDE</li> <li>– 2 - Server type is CDMA MPC</li> <li>– 3 - Server type is UMTS SLP</li> <li>– 4 - Server type is custom PDE</li> </ul> </li> </ul>
-------------------	---

<i>pIPv4Config</i>	<ul style="list-style-type: none"> <li>IPv4 address and port (Optional)</li> <li>See <a href="#">loc_IPv4Config</a> for more information</li> </ul>
<i>pIPv6Config</i>	<ul style="list-style-type: none"> <li>IPv6 address and port (Optional)</li> <li>See <a href="#">loc_IPv6Config</a> for more information</li> </ul>
<i>pURLAddr</i>	<ul style="list-style-type: none"> <li>Uniform Resource Locator (Optional)</li> <li>See <a href="#">loc_URLAddrInfo</a> for more information</li> </ul>

### 8.433.2 Field Documentation

8.433.2.1 `loc_IPv4Config* pack_loc_SLQSLOCSetServer_t::pIPv4Config`

8.433.2.2 `loc_IPv6Config* pack_loc_SLQSLOCSetServer_t::pIPv6Config`

8.433.2.3 `loc_URLAddrInfo* pack_loc_SLQSLOCSetServer_t::pURLAddr`

8.433.2.4 `uint32_t pack_loc_SLQSLOCSetServer_t::serverType`

## 8.434 `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`

### 8.434.1 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>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-AccuracyLvl</i>	<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>	<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>	<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>	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <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>• Optional Parameter</li> <li>• Configuration for Altitude Assumed Info in GNSS SV Info Event</li> <li>• Defaults to ENABLED.</li> <li>• Values <ul style="list-style-type: none"> <li>– 1 - Enable Altitude Assumed information in GNSS SV Info Event</li> <li>– 2 - Disable Altitude Assumed information in GNSS SV Info Event</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> </ul>

## 8.434.2 Field Documentation

8.434.2.1 `loc_LocApplicationInfo*` `pack_loc_Start_t::pApplicationInfo`

8.434.2.2 `uint32_t*` `pack_loc_Start_t::pConfigAltitudeAssumed`

8.434.2.3 `uint32_t*` `pack_loc_Start_t::pHorizontalAccuracyLvl`

8.434.2.4 `uint32_t*` `pack_loc_Start_t::pIntermediateReportState`

8.434.2.5 `uint32_t*` `pack_loc_Start_t::pMinIntervalTime`

8.434.2.6 uint32\_t\* pack\_loc\_Start\_t::pRecurrenceType

8.434.2.7 uint8\_t pack\_loc\_Start\_t::SessionId

8.434.2.8 uint16\_t pack\_loc\_Start\_t::Tlvresult

## 8.435 pack\_loc\_Stop\_t Struct Reference

### Data Fields

- uint8\_t [SessionId](#)
- uint16\_t [Tlvresult](#)

### 8.435.1 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>

### 8.435.2 Field Documentation

8.435.2.1 uint8\_t pack\_loc\_Stop\_t::SessionId

8.435.2.2 uint16\_t pack\_loc\_Stop\_t::Tlvresult

## 8.436 pack\_nas\_InitiateDomainAttach\_t Struct Reference

### Data Fields

- uint32\_t [action](#)

### 8.436.1 Detailed Description

Pack structure for Initiates a domain attach/detach of the device.

#### Parameters

<i>action</i>	<ul style="list-style-type: none"> <li>• Domain action to attempt               <ul style="list-style-type: none"> <li>1 - Attach</li> <li>2 - Detach</li> </ul> </li> </ul>
---------------	--

### 8.436.2 Field Documentation

8.436.2.1 uint32\_t pack\_nas\_InitiateDomainAttach\_t::action

## 8.437 pack\_nas\_PerformNetworkScanPCI\_t Struct Reference

### Data Fields

- uint8\_t \* [pNetworkType](#)
- uint32\_t \* [pScanType](#)
- uint64\_t \* [pCiotOpModePref](#)
- uint64\_t \* [pLteM1BandPref](#)
- uint64\_t \* [pLteNB1BandPref](#)

### 8.437.1 Detailed Description

Contain the network scan information.

#### Parameters

<i>pNetworkType</i>	<ul style="list-style-type: none"> <li>• network type.</li> <li>• Bit 0 - GSM.</li> <li>• Bit 1 - UMTS.</li> <li>• Bit 2 - LTE.</li> <li>• Bit 3 - TD-SCDMA.</li> </ul>
<i>pScanType</i>	<ul style="list-style-type: none"> <li>• Scan type.</li> <li>• 0x00 - PLMN(default).</li> <li>• 0x01 - Closed subscriber group.</li> <li>• 0x02 - Mode prference.</li> <li>• 0x03 - Physical cell ID.</li> </ul>
<i>pCiotOpMode-Pref[IN]</i>	<ul style="list-style-type: none"> <li>• CIOT LTE Operational Mode Preference</li> <li>• Bitmask representing the Cellular Internet of Things (CIoT) LTE operational mode preference.</li> <li>• Values: <ul style="list-style-type: none"> <li>– Bit 0 (0x01) - QMI_NAS_CIoT_LTE_OP_MODE_PREF_WB - LTE wideband (legacy LTE)</li> <li>– Bit 1 (0x02) - QMI_NAS_CIoT_LTE_OP_MODE_PREF_M1 - LTE M1</li> <li>– Bit 2 (0x04) - QMI_NAS_CIoT_LTE_OP_MODE_PREF_NB1 - LTE NB1</li> <li>– All unlisted bits are reserved for future use and the service point ignores them if used.</li> </ul> </li> </ul>

<p><i>pLteM1Band-Pref</i>[IN]</p>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE M1 band preferencet.</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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> </ul>
-----------------------------------	---

<p><i>pLteNb1Band-Pref</i>[IN]</p>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE NB1 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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> </ul>
------------------------------------	---

## 8.437.2 Field Documentation

### 8.437.2.1 uint64\_t\* pack\_nas\_PerformNetworkScanPCI\_t::pCiotOpModePref

8.437.2.2 uint64\_t\* pack\_nas\_PerformNetworkScanPCI\_t::pLteM1BandPref

8.437.2.3 uint64\_t\* pack\_nas\_PerformNetworkScanPCI\_t::pLteNB1BandPref

8.437.2.4 uint8\_t\* pack\_nas\_PerformNetworkScanPCI\_t::pNetworkType

8.437.2.5 uint32\_t\* pack\_nas\_PerformNetworkScanPCI\_t::pScanType

## 8.438 pack\_nas\_SetACCOLC\_t Struct Reference

### Data Fields

- int8\_t [spc](#) [6]
- uint8\_t [accolc](#)

### 8.438.1 Detailed Description

Sets the access overload class (ACCOLC)

#### Parameters

<i>spc</i>	<ul style="list-style-type: none"> <li>• service programming code NULL-terminated string of six digit</li> </ul>
<i>accolc</i>	<ul style="list-style-type: none"> <li>• ACCOLC : Valid range is 0 to 15</li> </ul>

### 8.438.2 Field Documentation

8.438.2.1 uint8\_t pack\_nas\_SetACCOLC\_t::accolc

8.438.2.2 int8\_t pack\_nas\_SetACCOLC\_t::spc[6]

## 8.439 pack\_nas\_SetCDMANetworkParameters\_t Struct Reference

### Data Fields

- char \* [pSPC](#)
- uint8\_t \* [pForceRev0](#)
- uint8\_t \* [pCustomSCP](#)
- uint32\_t \* [pProtocol](#)
- uint32\_t \* [pBroadcast](#)
- uint32\_t \* [pApplication](#)
- uint32\_t \* [pRoaming](#)

### 8.439.1 Detailed Description

Pack structure to set CDMA Network parameters.



## Parameters

<i>pSPC</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</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</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>	<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 Port</li> <li>– 0xFFFFFFFF - Unknown</li> </ul> </li> </ul>
<i>pBroadcast</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</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</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>

## 8.439.2 Field Documentation

8.439.2.1 uint32\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pApplication

8.439.2.2 uint32\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pBroadcast

8.439.2.3 uint8\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pCustomSCP

8.439.2.4 uint8\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pForceRev0

8.439.2.5 uint32\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pProtocol

8.439.2.6 uint32\_t\* pack\_nas\_SetCDMANetworkParameters\_t::pRoaming

8.439.2.7 char\* pack\_nas\_SetCDMANetworkParameters\_t::pSPC

## 8.440 pack\_nas\_SetNetworkPreference\_t Struct Reference

### Data Fields

- uint32\_t [TechnologyPref](#)
- uint32\_t [Duration](#)
- uint16\_t [Tlvresult](#)

### 8.440.1 Detailed Description

This structure contains pack set network preference parameters.

#### Parameters

<i>TechnologyPref</i> [IN]	<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</i> [IN]	<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>

### 8.440.2 Field Documentation

8.440.2.1 uint32\_t pack\_nas\_SetNetworkPreference\_t::Duration

8.440.2.2 uint32\_t pack\_nas\_SetNetworkPreference\_t::TechnologyPref

8.440.2.3 uint16\_t pack\_nas\_SetNetworkPreference\_t::Tlvresult

## 8.441 pack\_nas\_SLQSSConfigSigInfo\_t Struct Reference

### Data Fields

- [nas\\_RSSIThresh](#) \* [pRSSIThresh](#)
- [nas\\_ECIOThresh](#) \* [pECIOThresh](#)
- [nas\\_HDRSINRThresh](#) \* [pHDRSINRThresh](#)
- [nas\\_LTESNRThresh](#) \* [pLTESNRThresh](#)
- [nas\\_IOTresh](#) \* [pIOTresh](#)
- [nas\\_RSRQThresh](#) \* [pRSRQThresh](#)
- [nas\\_RSRPThresh](#) \* [pRSRPThresh](#)
- [nas\\_LTESigRptCfg](#) \* [pLTESigRptCfg](#)
- [nas\\_TDSCDMASINRCONFTresh](#) \* [pTDSCDMASINRCONFTresh](#)

### 8.441.1 Detailed Description

Structure for ConfigSigInfo pack.

#### Parameters

<i>pRSSIThresh</i>	<ul style="list-style-type: none"> <li>• RSSI threshold List</li> <li>• See <a href="#">nas_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="#">nas_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="#">nas_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="#">nas_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="#">nas_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="#">nas_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="#">nas_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="#">nas_LTESigRptCfg</a> for more details</li> </ul>
<i>pTDSCDMASINRCONFTthresh</i>	<ul style="list-style-type: none"> <li>• TD-SCDMA SINR Threshold List</li> <li>• See <a href="#">nas_TDSCDMASINRCONFTthresh</a> for more details</li> </ul>

### 8.441.2 Field Documentation

8.441.2.1 `nas_ECIOThresh* pack_nas_SLQSCfgSigInfo_t::pECIOThresh`

8.441.2.2 `nas_HDRSINRThresh* pack_nas_SLQSCfgSigInfo_t::pHDRSINRThresh`

8.441.2.3 `nas_IOTThresh* pack_nas_SLQSCfgSigInfo_t::pIOTThresh`

8.441.2.4 `nas_LTESigRptCfg* pack_nas_SLQSCfgSigInfo_t::pLTESigRptCfg`

8.441.2.5 `nas_LTESNRThresh* pack_nas_SLQSCfgSigInfo_t::pLTESNRThresh`

8.441.2.6 `nas_RSRPThresh* pack_nas_SLQSCfgSigInfo_t::pRSRPThresh`

8.441.2.7 `nas_RSRQThresh* pack_nas_SLQSCfgSigInfo_t::pRSRQThresh`

8.441.2.8 `nas_RSSIThresh* pack_nas_SLQSCfgSigInfo_t::pRSSIThresh`

8.441.2.9 `nas_TDSCDMASINRCONFTthresh* pack_nas_SLQSCfgSigInfo_t::pTDSCDMASINRCONFTthresh`

## 8.442 `pack_nas_SLQSGetPLMNName_t` Struct Reference

### Data Fields

- `uint16_t` [mcc](#)
- `uint16_t` [mnc](#)
- `uint8_t *` [pMncPcsStatus](#)

### 8.442.1 Detailed Description

This structure contains pack get operator name for specified network 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.

**8.442.2 Field Documentation**

8.442.2.1 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mcc

8.442.2.2 uint16\_t pack\_nas\_SLQSGetPLMNName\_t::mnc

8.442.2.3 uint8\_t\* pack\_nas\_SLQSGetPLMNName\_t::pMncPcsStatus

**8.443 pack\_nas\_SLQSIInitiateNetworkRegistration\_t Struct Reference****Data Fields**

- uint32\_t [regAction](#)
- [nas\\_MNRInfo](#) \* [pMNRInfo](#)
- uint32\_t \* [pChangeDuration](#)
- uint8\_t \* [pMncPcsDigitStatus](#)

**8.443.1 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 MNRInfo <ul style="list-style-type: none"> <li>– See <a href="#">nas_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>

## 8.443.2 Field Documentation

8.443.2.1 `uint32_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pChangeDuration`

8.443.2.2 `uint8_t* pack_nas_SLQSIInitiateNetworkRegistration_t::pMncPcsDigitStatus`

8.443.2.3 `nas_MNRInfo* pack_nas_SLQSIInitiateNetworkRegistration_t::pMNRInfo`

8.443.2.4 `uint32_t pack_nas_SLQSIInitiateNetworkRegistration_t::regAction`

## 8.444 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 * pHDRSINRThreshold`
- `uint16_t * pHDRSINRDelta`
- `nas_HDRIOThresh * pHDRIOThresh`
- `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 * pLTESNRThreshold`
- `uint16_t * pLTESNRDelta`
- `nas_LTERSRQThresh * pLTERSRQThresh`
- `uint16_t * pLTERSRQDelta`
- `nas_LTERSRPThresh * pLTERSRPThresh`
- `uint16_t * pLTERSRPDelta`
- `nas_LTESigRptConfig * pLTERSigRptConfig`

- [nas\\_TDSCDMARSCPThresh](#) \* [pTDSCDMARSCPThresh](#)
- [uint16\\_t](#) \* [pTDSCDMARSCPDelta](#)
- [nas\\_TDSCDMARSSIThresh](#) \* [pTDSCDMARSSIThresh](#)
- [float](#) \* [pTDSCDMARSSIDelta](#)
- [nas\\_TDSCDMAECIOThresh](#) \* [pTDSCDMAECIOThresh](#)
- [float](#) \* [pTDSCDMAECIODelta](#)
- [nas\\_TDSCDMASINRThresh](#) \* [pTDSCDMASINRThresh](#)
- [float](#) \* [pTDSCDMASINRDelta](#)

### 8.444.1 Detailed Description

This structure contains pack set the signal strength reporting thresholds parameters.

#### 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>pHDRIOThresh</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="#">nas_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="#">nas_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>pLTESigRpt-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>

**Note**

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.

**8.444.2 Field Documentation**

8.444.2.1 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMAECIODelta

8.444.2.2 nas\_CDMAECIOThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMAECIOThresh

8.444.2.3 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMARSSIDelta

8.444.2.4 nas\_CDMARSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pCDMARSSIThresh

8.444.2.5 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pGSMRSSIDelta

8.444.2.6 nas\_GSMRSSIThresh\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pGSMRSSIThresh

8.444.2.7 uint16\_t\* pack\_nas\_SLQSNasConfigSigInfo2\_t::pHDRECIODelta

- 8.444.2.8 `nas_HDRECIOTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pHDRECIOTresh`
- 8.444.2.9 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pHDRIODelta`
- 8.444.2.10 `nas_HDRIOTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pHDRIOTresh`
- 8.444.2.11 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSIDelta`
- 8.444.2.12 `nas_HDRRSSITresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pHRRSSITresh`
- 8.444.2.13 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRDelta`
- 8.444.2.14 `nas_HDRSINRThreshold*` `pack_nas_SLQSNasConfigSigInfo2_t::pHRSINRThresh`
- 8.444.2.15 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPDelta`
- 8.444.2.16 `nas_LTERSRPTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRPTresh`
- 8.444.2.17 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQDelta`
- 8.444.2.18 `nas_LTERSRQTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSRQTresh`
- 8.444.2.19 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSIDelta`
- 8.444.2.20 `nas_LTERSSITresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTERSSITresh`
- 8.444.2.21 `nas_LTESigRptConfig*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESigRptConfig`
- 8.444.2.22 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRDelta`
- 8.444.2.23 `nas_LTESNRThreshold*` `pack_nas_SLQSNasConfigSigInfo2_t::pLTESNRThresh`
- 8.444.2.24 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIODelta`
- 8.444.2.25 `nas_TDSCDMAECIOTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMAECIOTresh`
- 8.444.2.26 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPDelta`
- 8.444.2.27 `nas_TDSCDMARSCPThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSCPThresh`
- 8.444.2.28 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSIDelta`
- 8.444.2.29 `nas_TDSCDMARSSITresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMARSSITresh`
- 8.444.2.30 `float*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRDelta`
- 8.444.2.31 `nas_TDSCDMASINRThresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pTDSCDMASINRThresh`
- 8.444.2.32 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIODelta`
- 8.444.2.33 `nas_WCMAECIOTresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCMAECIOTresh`
- 8.444.2.34 `uint16_t*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSIDelta`
- 8.444.2.35 `nas_WCDMARSSITresh*` `pack_nas_SLQSNasConfigSigInfo2_t::pWCDMARSSITresh`

## 8.445 pack\_nas\_SLQSNasGet3GPP2Subscription\_t Struct Reference

### Data Fields

- uint8\_t [namID](#)

#### 8.445.1 Detailed Description

Structure for 3GPP2 Subscription pack.

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

#### 8.445.2 Field Documentation

8.445.2.1 uint8\_t pack\_nas\_SLQSNasGet3GPP2Subscription\_t::namID

## 8.446 pack\_nas\_SLQSNASGeteDRXParamsExt\_t Struct Reference

### Data Fields

- uint8\_t \* [pEdrxRAT](#)
- uint32\_t \* [pLteOpMode](#)

#### 8.446.1 Detailed Description

This structure contains the SLQSNASGeteDRXParamsExt request parameters.

##### Parameters

<i>pEdrxRAT</i>	[Optional] <ul style="list-style-type: none"><li>eDRX Radio Access Technology</li><li>Values<ul style="list-style-type: none"><li>NAS_RADIO_IF_NO_SVC (0x00) - None (no service)</li><li>NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X</li><li>NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO)</li><li>NAS_RADIO_IF_AMPS (0x03) - AMPS</li><li>NAS_RADIO_IF_GSM (0x04) - GSM</li><li>NAS_RADIO_IF_UMTS (0x05) - UMTS</li><li>NAS_RADIO_IF_WLAN (0x06) - WLAN</li><li>NAS_RADIO_IF_GPS (0x07) - GPS</li><li>NAS_RADIO_IF_LTE (0x08) - LTE</li><li>NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA</li><li>NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1</li><li>NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1</li><li>NAS_RADIO_IF_NO_CHANGE (-1) - No change Note: The device will use "0x08 - NAS_RADIO_IF_LTE" as the default value if the TLV is omitted.</li></ul></li></ul>
-----------------	---

<i>pLteOpMode</i>	[Optional] <ul style="list-style-type: none"> <li>• LTE Operational Mode</li> <li>• Values             <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1 Note: This TLV should be set when edrx_rat_type is LTE.</li> </ul> </li> </ul>
-------------------	---

## 8.446.2 Field Documentation

8.446.2.1 uint8\_t\* pack\_nas\_SLQSNASGeteDRXParamsExt\_t::pEdrxRAT

8.446.2.2 uint32\_t\* pack\_nas\_SLQSNASGeteDRXParamsExt\_t::pLteOpMode

## 8.447 pack\_nas\_SLQSNasGetTxRxInfo\_t Struct Reference

### Data Fields

- uint8\_t [radio\\_if](#)

### 8.447.1 Detailed Description

Structure for TxRx Info pack

#### 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_1XEVDO - 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>
-----------------	---

## 8.447.2 Field Documentation

8.447.2.1 uint8\_t pack\_nas\_SLQSNasGetTxRxInfo\_t::radio\_if

## 8.448 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 \* [pNetworkRejectInd](#)
- uint8\_t \* [pSuppressSysInfoInd](#)
- uint8\_t \* [pLTECphyCa](#)
- uint8\_t \* [pEdrxChangeInfoInd](#)

### 8.448.1 Detailed Description

This structure contains pack indication register parameters.

#### Parameters

<i>pSystem- SelectionInd</i>	[Optional] <ul style="list-style-type: none"> <li>• System Selection Preference indication registration. The 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>
<i>pDDTMInd</i>	[Optional] <ul style="list-style-type: none"> <li>• DDTM (Data Dedicated Transmission Mode) indication registration. The 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>
<i>pServing- SystemInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Serving System indication registration. The 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>
<i>pDualStandBy- PrefInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Dual Standby Preference indication registration. The 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>
<i>pSubscription- InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Subscription Information indication registration. The 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>
<i>pNetworkTime- Ind</i>	[Optional] <ul style="list-style-type: none"> <li>• Network Time indication registration. The 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>

<i>pSysInfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>System Information indication registration. The 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>
<i>pSignalStrength-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>Signal Strength indication registration. The 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>
<i>pErrorRateInd</i>	[Optional] <ul style="list-style-type: none"> <li>Error Rate indication registration. The 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>
<i>pHDRNewUATI-AssInd</i>	[Optional] <ul style="list-style-type: none"> <li>HDR New UATI Assigned indication registration. The 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>
<i>pHDRSession-CloseInd</i>	[Optional] <ul style="list-style-type: none"> <li>HDR Session Closed indication registration. The 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>
<i>pManaged-RoamingInd</i>	[Optional] <ul style="list-style-type: none"> <li>Managed Roaming indication registration. The callbacks would not be invoked if the indication is disabled.             <ul style="list-style-type: none"> <li>0x00 - Disable</li> </ul> </li> </ul>
<i>pNetworkReject-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>Network reject information 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>
<i>pSuppressSys-InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values:             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable- 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.             <ul style="list-style-type: none"> <li>0x00 - Disable (default value)</li> <li>0x01 - Enable</li> </ul> </li> </ul>

<i>pEdrxChange-InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>EDRX Change 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>
----------------------------	---

## 8.448.2 Field Documentation

- 8.448.2.1 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDDTMInd`
- 8.448.2.2 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pDualStandByPrefInd`
- 8.448.2.3 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pEdrxChangeInfoInd`
- 8.448.2.4 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pErrorRateInd`
- 8.448.2.5 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRNewUATIAssInd`
- 8.448.2.6 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pHDRSessionCloseInd`
- 8.448.2.7 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pLTECphyCa`
- 8.448.2.8 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pManagedRoamingInd`
- 8.448.2.9 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkRejectInd`
- 8.448.2.10 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pNetworkTimeInd`
- 8.448.2.11 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pServingSystemInd`
- 8.448.2.12 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSignalStrengthInd`
- 8.448.2.13 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSubscriptionInfoInd`
- 8.448.2.14 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSuppressSysInfoInd`
- 8.448.2.15 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSysInfoInd`
- 8.448.2.16 `uint8_t* pack_nas_SLQSNasIndicationRegisterExt_t::pSystemSelectionInd`

## 8.449 pack\_nas\_SLQSNasIndicationRegisterV2\_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 \* [pCurrentPLMNNameInd](#)
- uint8\_t \* [pEMBMSStatusInd](#)
- uint8\_t \* [pRFBandInfoInd](#)
- uint8\_t \* [pNetworkRejectInd](#)
- uint8\_t \* [pSuppressSysInfoInd](#)
- uint8\_t \* [pOperatorNameDataInd](#)
- uint8\_t \* [pCSPPLMNModelInd](#)
- uint8\_t \* [pRTREConfigurationInd](#)
- uint8\_t \* [pIMSPrefStatusInd](#)
- uint8\_t \* [pE911StateReadyStatusInd](#)
- uint8\_t \* [pLTSIB16NetworkTimeInd](#)
- uint8\_t \* [pLTECphyCa](#)
- uint8\_t \* [pSubscriptionChangeInd](#)
- uint8\_t \* [pSSACBarringInd](#)
- uint8\_t \* [pT3402TimerValueInd](#)
- uint8\_t \* [pAccessClassBarringInd](#)
- uint8\_t \* [pDataSubscriptionPriorityInd](#)
- uint8\_t \* [pT3346TimerStatusChangeInd](#)
- uint8\_t \* [pCallModeStatusInd](#)
- uint8\_t \* [pSSACBarringExtInd](#)
- uint8\_t \* [pManualNWScanFailureInd](#)
- uint8\_t \* [pTimerExpiryInd](#)
- uint8\_t \* [pEmergencyModeStatusInd](#)
- uint8\_t \* [pGcellInfoInd](#)
- uint8\_t \* [pEdrxChangeInfoInd](#)
- uint8\_t \* [pLTERACHFailInd](#)
- uint8\_t \* [pLTERRCTxInfoInd](#)

### 8.449.1 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.                tFNRoamingIndicator tFNDataCapabilities and tFNServingSystem               <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.                tFNDDTM               <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. tFNBandPreference             <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. tFNDualStandByPref             <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. tFNSubscriptionInfo             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pNetworkTime-Ind</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. tFNNetworkTime             <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. tFNSysInfo             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSignalStrength-Ind</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. tFNSigInfo             <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. tFNErrRate             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pHDRNewUATI-AssInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• HDR New UATI Assigned indication registration. The following callbacks would not be invoked if the indication is disabled. tFNHDRUATIUpdate             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pHDRSession-CloseInd</i>	<p>[Optional]</p> <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>	<p>[Optional]</p> <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>pCurrentPLMN-NameInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Current PLMN Name indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_CURRENT_PLMN_NAME_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pEMBMSStatus-Ind</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• eMBMS Status indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_EMBMS_STATUS_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pRFBandInfoInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• RF Band Information indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_RF_BAND_INFO_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pNetworkReject-Ind</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Network reject information indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>

<i>pSuppressSys-InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>Controls the reporting of QMI_NAS_SYS_INFO_IND when only the reject_cause field has changed. Values:             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pOperatorName-DataInd</i>	[Optional] <ul style="list-style-type: none"> <li>Operator Name Data indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_OPERATOR_NAME_DATA_IND             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pCSPPLMN-ModeInd</i>	[Optional] <ul style="list-style-type: none"> <li>CSP PLMN Mode Bit indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_CSP_PLMN_MODE_BIT_IND             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pRTRE-ConfigurationInd</i>	[Optional] <ul style="list-style-type: none"> <li>RTRE Configuration indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_RTRE_CONFIG_IND             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pIMSPrefStatus-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>IMS Preference Status indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_IMS_PREF_STATUS_IND             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pE911State-ReadyStatusInd</i>	[Optional] <ul style="list-style-type: none"> <li>E911 State Ready Status indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_E911_STATE_READY_IND             <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pLTSIB16-NetworkTimeInd</i>	[Optional] <ul style="list-style-type: none"> <li>LTE SIB16 Network Time indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>QMI_NAS_LTE_SIB16_NETWORK_TIME_IND             <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 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>
<i>pSubscription-ChangeInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Subscription Change indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_SUBSCRIPTION_CHANGE_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSSACBarring-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>• Service-Specific Access Class Barring indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_SSAC_INFO_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pT3402Timer-ValueInd</i>	[Optional] <ul style="list-style-type: none"> <li>• T3402 Timer Value indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_EMM_T3402_CHANGED_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pAccessClass-BarringInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Access Class Barring indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_ACB_INFO_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pData-Subscription-PriorityInd</i>	[Optional] <ul style="list-style-type: none"> <li>• Data Subscription Priority indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_DATA_SUBS_PRIORITY_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pT3346Timer-StatusChange-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>• T3346 Timer Status Change indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_DATA_SUBS_PRIORITY_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

<i>pCallMode-StatusInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Call Mode Status indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_CALL_MODE_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pSSACBarring-ExtInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Service-Specific Access Class Barring Ext indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_SSAC_CHANGE_INFO_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pManualNW-ScanFailureInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Manual Network Scan Failure indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_MANUAL_SCAN_FAIL_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pTimerExpiryInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Timer Expiry indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_TIMER_EXPIRY_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pEmergency-ModeStatusInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Emergency Mode Status indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_EMERGENCY_MODE_STATUS_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pGcellInfoInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• Gcell Info indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_GCELL_INFO_IND <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pEdrxChange-InfoInd</i>	<p>[Optional]</p> <ul style="list-style-type: none"> <li>• EDRX Change Info indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul>

<i>pLTERACHFail-Ind</i>	[Optional] <ul style="list-style-type: none"> <li>• LTE RACH Failure Indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_LTE_RACH_FAIL_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>pLTERRCTx-InfoInd</i>	[Optional] <ul style="list-style-type: none"> <li>• LTE RRC Tx Info Indication registration. The following callbacks would not be invoked if the indication is disabled.</li> <li>• QMI_NAS_LTE_RRC_TX_INFO_IND             <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

**Note**

Atleast one parameter must be provided as request. 'NULL' value confirms that the indication value is not sent.

**8.449.2 Field Documentation**

- 8.449.2.1 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pAccessClassBarringInd
- 8.449.2.2 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pCallModeStatusInd
- 8.449.2.3 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pCSPPLMNModelInd
- 8.449.2.4 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pCurrentPLMNNameInd
- 8.449.2.5 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pDataSubscriptionPriorityInd
- 8.449.2.6 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pDDTMInd
- 8.449.2.7 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pDualStandByPrefInd
- 8.449.2.8 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pE911StateReadyStatusInd
- 8.449.2.9 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pEdrxChangeInfoInd
- 8.449.2.10 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pEMBMSStatusInd
- 8.449.2.11 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pEmergencyModeStatusInd
- 8.449.2.12 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pErrorRateInd
- 8.449.2.13 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pGcellInfoInd
- 8.449.2.14 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pHDRNewUATIAssInd
- 8.449.2.15 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pHDRSessionCloseInd
- 8.449.2.16 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pIMSPrefStatusInd

- 8.449.2.17 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pLTECphyCa
- 8.449.2.18 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pLTERACHFailInd
- 8.449.2.19 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pLTERRCTxInfoInd
- 8.449.2.20 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pLTESIB16NetworkTimeInd
- 8.449.2.21 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pManagedRoamingInd
- 8.449.2.22 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pManualNWScanFailureInd
- 8.449.2.23 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pNetworkRejectInd
- 8.449.2.24 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pNetworkTimeInd
- 8.449.2.25 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pOperatorNameDataInd
- 8.449.2.26 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pRFBandInfoInd
- 8.449.2.27 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pRTREConfigurationInd
- 8.449.2.28 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pServingSystemInd
- 8.449.2.29 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSignalStrengthInd
- 8.449.2.30 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSSACBarringExtInd
- 8.449.2.31 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSSACBarringInd
- 8.449.2.32 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSubscriptionChangeInd
- 8.449.2.33 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSubscriptionInfoInd
- 8.449.2.34 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSuppressSysInfoInd
- 8.449.2.35 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSysInfoInd
- 8.449.2.36 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pSystemSelectionInd
- 8.449.2.37 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pT3346TimerStatusChangeInd
- 8.449.2.38 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pT3402TimerValueInd
- 8.449.2.39 uint8\_t\* pack\_nas\_SLQSNasIndicationRegisterV2\_t::pTimerExpiryInd

## 8.450 pack\_nas\_SLQSNASSeteDRXParams\_t Struct Reference

### Data Fields

- uint8\_t \* [pEdrxEnable](#)
- uint8\_t \* [pCycleLen](#)
- uint8\_t \* [pPagingTimeWindow](#)
- uint8\_t \* [pEdrxRatType](#)
- uint32\_t \* [pEdrxCiotLteMode](#)

### 8.450.1 Detailed Description

This structure contains the SLQSNASSeteDRXParams request parameters.

#### Parameters

<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> <li>eDRX Enable</li> <li>Values               <ul style="list-style-type: none"> <li>0x00 - Disable</li> <li>0x01 - Enable</li> </ul> </li> </ul>
<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> <li>eDRX cycle length</li> <li>Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> </ul>
<i>pPagingTimeWindow</i>	[Optional] <ul style="list-style-type: none"> <li>eDRX paging time window</li> <li>Value to be set per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> </ul>
<i>pEdrxRatType</i>	[Optional] <ul style="list-style-type: none"> <li>Radio access technology</li> <li>Values               <ul style="list-style-type: none"> <li>NAS_RADIO_IF_NO_SVC (0x00) - None (no service)</li> <li>NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X</li> <li>NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO)</li> <li>NAS_RADIO_IF_AMPS (0x03) - AMPS</li> <li>NAS_RADIO_IF_GSM (0x04) - GSM</li> <li>NAS_RADIO_IF_UMTS (0x05) - UMTS</li> <li>NAS_RADIO_IF_WLAN (0x06) - WLAN</li> <li>NAS_RADIO_IF_GPS (0x07) - GPS</li> <li>NAS_RADIO_IF_LTE (0x08) - LTE</li> <li>NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA</li> <li>NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1</li> <li>NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1</li> <li>NAS_RADIO_IF_NO_CHANGE (-1) - No change Note: The device will use "0x08 - NAS_RADIO_IF_LTE" as the default value if the TLV is omitted.</li> </ul> </li> </ul>
<i>pEdrxCiotLteMode</i>	[Optional] <ul style="list-style-type: none"> <li>CIOT LTE mode</li> <li>Values               <ul style="list-style-type: none"> <li>NAS_CIoT_SYS_MODE_NO_SRV(0x00) - No service</li> <li>NAS_CIoT_SYS_MODE_LTE_WB(0x01) - Camped on LTE wideband</li> <li>NAS_CIoT_SYS_MODE_LTE_M1(0x02) - Camped on LTE M1</li> <li>NAS_CIoT_SYS_MODE_LTE_NB1(0x03) - Camped on LTE NB1 Note: This TLV should be set when pEdrxRatType is LTE.</li> </ul> </li> </ul>

### 8.450.2 Field Documentation

#### 8.450.2.1 uint8\_t\* pack\_nas\_SLQSNASSeteDRXParams\_t::pCycleLen



8.450.2.2 uint32\_t\* pack\_nas\_SLQSNASSeteDRXParams\_t::pEdrxCiotLteMode

8.450.2.3 uint8\_t\* pack\_nas\_SLQSNASSeteDRXParams\_t::pEdrxEnable

8.450.2.4 uint8\_t\* pack\_nas\_SLQSNASSeteDRXParams\_t::pEdrxRatType

8.450.2.5 uint8\_t\* pack\_nas\_SLQSNASSeteDRXParams\_t::pPagingTimeWindow

## 8.451 pack\_nas\_SLQSNasSwIndicationRegister\_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](#)
- uint8\_t \* [pTimer](#)

### 8.451.1 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>

<i>pTimer</i>	<ul style="list-style-type: none"> <li>• 0 - do not report</li> <li>• 1 - report Timer Indicator messages</li> </ul>
---------------	--

## 8.451.2 Field Documentation

8.451.2.1 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::gsmUmtsDI

8.451.2.2 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::gsmUmtsUI

8.451.2.3 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::lteEmmDI

8.451.2.4 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::lteEmmUI

8.451.2.5 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::lteEsmDI

8.451.2.6 uint8\_t pack\_nas\_SLQSNasSwiIndicationRegister\_t::lteEsmUI

8.451.2.7 uint8\_t\* pack\_nas\_SLQSNasSwiIndicationRegister\_t::pRankIndicatorInd

8.451.2.8 uint8\_t\* pack\_nas\_SLQSNasSwiIndicationRegister\_t::pTimer

## 8.452 pack\_nas\_SLQSNASSwiSetChannelLock\_t Struct Reference

### Data Fields

- [nas\\_wcdmaUARFCN](#) \* [pWcdmaUARFCN](#)
- [nas\\_lteEARFCN](#) \* [pLteEARFCN](#)
- [nas\\_ltePCI](#) \* [pLtePCI](#)

### 8.452.1 Detailed Description

Structure for ChannelLock pack.

#### Parameters

<i>pWcdmaUARFCN</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_wcdmaUARFCN</a> for more information</li> </ul>
<i>pLteEARFCN</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_lteEARFCN</a> for more information</li> </ul>
<i>pLtePCI</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_ltePCI</a> for more information</li> </ul>

## 8.452.2 Field Documentation

8.452.2.1 nas\_lteEARFCN\* pack\_nas\_SLQSNASSwiSetChannelLock\_t::pLteEARFCN

8.452.2.2 nas\_ltePCI\* pack\_nas\_SLQSNASSwiSetChannelLock\_t::pLtePCI

8.452.2.3 nas\_wcdmaUARFCN\* pack\_nas\_SLQSNASSwiSetChannelLock\_t::pWcdmaUARFCN

## 8.453 pack\_nas\_SLQSSetBandPreference\_t Struct Reference

### Data Fields

- uint64\_t [bandPref](#)

### 8.453.1 Detailed Description

Provides information about the band preference.

### Parameters

---

<i>bandpreference</i>	<ul style="list-style-type: none"> <li>• Bit mask representing the band preference to be set.</li> <li>• Bit position meanings: <ul style="list-style-type: none"> <li>– 0 - BC0_A - Band Class 0, A-System</li> <li>– 1 - BC0_B - Band Class 0, B-System, Band Class 0 AB , GSM 850 Band</li> <li>– 2 - BC1 - Band Class 1, all blocks</li> <li>– 3 - BC2 - Band Class 2 place holder</li> <li>– 4 - BC3 - Band Class 3, A-System</li> <li>– 5 - BC4 - Band Class 4, all blocks</li> <li>– 6 - BC5 - Band Class 5, all blocks</li> <li>– 7 - GSM_DCS_1800 - GSM DCS band</li> <li>– 8 - GSM_EGSM_900 - GSM Extended GSM (E-GSM) band</li> <li>– 9 - GSM_PGSM_900 - GSM Primary GSM (P-GSM) band</li> <li>– 10 - BC6 - Band Class 6</li> <li>– 11 - BC7 - Band Class 7</li> <li>– 12 - BC8 - Band Class 8</li> <li>– 13 - BC9 - Band Class 9</li> <li>– 14 - BC10 - Band Class 10</li> <li>– 15 - BC11 - Band Class 11</li> <li>– 16 - GSM_450 - GSM 450 band</li> <li>– 17 - GSM_480 - GSM 480 band</li> <li>– 18 - GSM_750 - GSM 750 band</li> <li>– 19 - GSM_850 - GSM 850 band</li> <li>– 20 - GSM_RGSM_900 - GSM Railways GSM Band</li> <li>– 21 - GSM_PCS_1900 - GSM PCS band</li> <li>– 22 - WCDMA_I_IMT_2000 - WCDMA EUROPE JAPAN and CHINA IMT 2100 band</li> <li>– 23 - WCDMA_II_PCS_1900 - WCDMA US PCS 1900 band</li> <li>– 24 - WCDMA_III_1700 - WCDMA EUROPE and CHINA DCS 1800 band</li> <li>– 25 - WCDMA_IV_1700 - WCDMA US 1700 band</li> <li>– 26 - WCDMA_V_850 - WCDMA US 850 band</li> <li>– 27 - WCDMA_VI_800 - WCDMA JAPAN 800 band</li> <li>– 28 - BC12 - Band Class 12</li> <li>– 29 - BC14 - Band Class 14</li> <li>– 30 - RESERVED_2 - Reserved 2</li> <li>– 31 - BC15 - Band Class 15</li> <li>– 32 - 47 - Reserved</li> <li>– 48 - WCDMA_VII_2600 - WCDMA EUROPE 2600 band</li> <li>– 49 - WCDMA_VIII_900 - WCDMA EUROPE and JAPAN 900 band</li> <li>– 50 - WCDMA_IX_1700 - WCDMA JAPAN 1700 band</li> <li>– 51 to 55 - Reserved</li> <li>– 56 - BBC16 - Band Class 16</li> <li>– 57 - BC17 - Band Class 17</li> <li>– 58 - BC18 - Band Class 18</li> <li>– 59 - BC19 - Band Class 19</li> <li>– 60 to 64 - Reserved</li> </ul> </li> </ul>
-----------------------	---

## 8.453.2 Field Documentation

8.453.2.1 uint64\_t pack\_nas\_SLQSSetBandPreference\_t::bandPref

## 8.454 pack\_nas\_SLQSSetSignalStrengthsCallback\_t Struct Reference

### Data Fields

- uint8\_t [bEnable](#)
- [nas\\_SLQSSignalStrengthsIndReq](#) \* [pSigIndReq](#)

### 8.454.1 Detailed Description

This structure contains paramaters pack set strength thresholds callback.

#### Parameters

<i>bEnable</i>	0/1 to disable/enable RSSI signal streghth indication
<i>pSigIndReq</i>	parameters to control signal strength indication

### 8.454.2 Field Documentation

8.454.2.1 uint8\_t pack\_nas\_SLQSSetSignalStrengthsCallback\_t::bEnable

8.454.2.2 [nas\\_SLQSSignalStrengthsIndReq](#)\* [pack\\_nas\\_SLQSSetSignalStrengthsCallback\\_t::pSigIndReq](#)

## 8.455 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](#)

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

## 8.455.2 Field Documentation

8.455.2.1 struct nas\_acqOrderPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pAcqOrderPref

8.455.2.2 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pBandPref

8.455.2.3 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pChgDuration

8.455.2.4 struct nas\_CSGID\* pack\_nas\_SLQSSetSysSelectionPref\_t::pCSGID

8.455.2.5 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pEmerMode

8.455.2.6 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pGWAcqOrderPref

8.455.2.7 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pLTEBandPref

8.455.2.8 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pMNCIncPCSDigStat

8.455.2.9 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pModePref

8.455.2.10 struct nas\_netSelectionPref\* pack\_nas\_SLQSSetSysSelectionPref\_t::pNetSelPref

8.455.2.11 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pPRLPref

8.455.2.12 unsigned char\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRAT

8.455.2.13 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pRoamPref

8.455.2.14 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvDomainPref

8.455.2.15 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pSrvRegRestriction

8.455.2.16 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPref\_t::pTdsdmaBandPref

## 8.456 pack\_nas\_SLQSSetSysSelectionPrefExt\_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
- struct nas\_lteBandPrefExt \* pLTEBandPrefExt
- uint64\_t \* pCiotLteOpModePref
- uint64\_t \* pLteM1BandPref
- uint64\_t \* pLteNb1BandPref
- uint32\_t \* pCiotLteOpMode
- struct nas\_ciotAcqOrderPref \* pCiotAcqOrderPref
- struct nas\_nr5gBandPref \* pNr5gBandPref

### 8.456.1 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><li>– Bit 5 - TD-SCDMA</li><li>– Bit 6 - NR5G</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>- netSelectionPref</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="#">nas_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>- acqOrderPref</li> <li>• Optional parameter for specifying Acquisition Order Preference</li> <li>• see <a href="#">nas_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>- CSGID</li> <li>• Optional parameter for specifying CSG ID</li> <li>• Either of pNetSelPref or pCSGID can be set.</li> <li>• see <a href="#">nas_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> <li>– 0x0A - LTE-M1</li> <li>– 0x0B - LTE-NB1</li> <li>– 0x0C - NR5G</li> </ul> </li> </ul>
<i>pLTEBandPref-Ext[IN]</i>	<ul style="list-style-type: none"> <li>• LTE Band Preference Extended</li> <li>• see <a href="#">nas_lteBandPrefExt</a> for more information</li> </ul>
<i>pCiotLteOp-ModePref[IN]</i>	<ul style="list-style-type: none"> <li>• CIOT LTE Operational Mode Preference</li> <li>• Bitmask representing the ClotT LTE operational mode preference to be set.</li> <li>• Values : <ul style="list-style-type: none"> <li>– Bit 0 - LTE wideband</li> <li>– Bit 1 - LTE M1</li> <li>– Bit 2 - LTE NB1 All unlisted bits are reserved for future use and the service point ignores them if used. If this TLV is not present, the modem will use the value read from the NV during bootup.</li> </ul> </li> </ul>

<p><i>pLteM1Band-Pref</i>[IN]</p>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE M1 band preference to be set.</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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> </ul>
-----------------------------------	--

- If this field is not present, the modem will use the M1 band preference value read from the NV.

## Parameters

<p><i>pLteNb1Band-Pref</i>[IN]</p>	<ul style="list-style-type: none"> <li>• Bitmask representing the LTE NB1 band preference to be set.</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>– Bit 61 - E-UTRA Operating Band 126</li> <li>– Bit 62 - E-UTRA Operating Band 127</li> </ul> </li> <li>• If this field is not present, the modem will use the NB1 band preference value read from the NV.</li> </ul>
------------------------------------	---

<i>pCiotLteOpMode</i>	<ul style="list-style-type: none"> <li>• This TLV is used when the net_sel_pref is MANUAL and RAT TLV is LTE.</li> <li>• This TLV indicates the the operational mode to be used during the manual LTE PLMN selection.</li> <li>• Values : <ul style="list-style-type: none"> <li>– 0x00 - No service</li> <li>– 0x01 - Camped on LTE wideband</li> <li>– 0x02 - Camped on LTE M1</li> <li>– 0x03 - Camped on LTE NB1</li> </ul> </li> </ul>
<i>pCiotAcqOrderPref</i> [IN]	<ul style="list-style-type: none"> <li>• CIOT Acquisition Order Preference</li> <li>• see <a href="#">nas_ciotAcqOrderPref</a> for more information</li> </ul>
<i>pNr5gBandPref</i> [IN]	<ul style="list-style-type: none"> <li>• NR5G Band Preference</li> <li>• see <a href="#">nas_nr5gBandPref</a> for more information</li> </ul>

## 8.456.2 Field Documentation

- 8.456.2.1 struct nas\_acqOrderPref\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pAcqOrderPref
- 8.456.2.2 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pBandPref
- 8.456.2.3 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pChgDuration
- 8.456.2.4 struct nas\_ciotAcqOrderPref\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pCiotAcqOrderPref
- 8.456.2.5 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pCiotLteOpMode
- 8.456.2.6 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pCiotLteOpModePref
- 8.456.2.7 struct nas\_CSGID\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pCSGID
- 8.456.2.8 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pEmerMode
- 8.456.2.9 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pGWAcqOrderPref
- 8.456.2.10 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pLTEBandPref
- 8.456.2.11 struct nas\_lteBandPrefExt\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pLTEBandPrefExt
- 8.456.2.12 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pLteM1BandPref
- 8.456.2.13 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pLteNb1BandPref
- 8.456.2.14 uint8\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pMNCIncPCSDigStat
- 8.456.2.15 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pModePref
- 8.456.2.16 struct nas\_netSelectionPref\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pNetSelPref
- 8.456.2.17 struct nas\_nr5gBandPref\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pNr5gBandPref

8.456.2.18 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pPRLPref

8.456.2.19 unsigned char\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pRAT

8.456.2.20 uint16\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pRoamPref

8.456.2.21 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pSrvDomainPref

8.456.2.22 uint32\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pSrvRegRestriction

8.456.2.23 uint64\_t\* pack\_nas\_SLQSSetSysSelectionPrefExt\_t::pTdsdmaBandPref

## 8.457 pack\_nas\_SLQSSwiPSDetach\_t Struct Reference

### Data Fields

- uint8\_t \* [pDetachAction](#)

### 8.457.1 Detailed Description

Structure for PS connection detach pack.

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

### 8.457.2 Field Documentation

8.457.2.1 uint8\_t\* pack\_nas\_SLQSSwiPSDetach\_t::pDetachAction

## 8.458 pack\_pds\_PDSInjectTimeReference\_t Struct Reference

### Data Fields

- uint64\_t [systemTime](#)
- uint16\_t [systemDiscontinuities](#)

### 8.458.1 Detailed Description

Structure contain the parameter for PDS Inject time.

#### Parameters

<i>systemTime</i>	<ul style="list-style-type: none"> <li>• System time( milliseconds )</li> </ul>
<i>system-Discontinuities</i>	<ul style="list-style-type: none"> <li>• Number of system time discontinuities</li> </ul>

## 8.458.2 Field Documentation

8.458.2.1 uint16\_t pack\_pds\_PDSInjectTimeReference\_t::systemDiscontinuities

8.458.2.2 uint64\_t pack\_pds\_PDSInjectTimeReference\_t::systemTime

## 8.459 pack\_pds\_ResetPDSData\_t Struct Reference

### Data Fields

- uint32\_t \* [pGPSDataMask](#)
- uint32\_t \* [pCellDataMask](#)

### 8.459.1 Detailed Description

Structure contain the parameter to Resets the specified PDS data.

#### Parameters

<i>pGPSDataMask</i>	<ul style="list-style-type: none"> <li>• Bitmask of GPS data to clear (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - EPH</li> <li>– 0x00000002 - ALM</li> <li>– 0x00000004 - POS</li> <li>– 0x00000008 - TIME</li> <li>– 0x00000010 - IONO</li> <li>– 0x00000020 - UTC</li> <li>– 0x00000040 - HEALTH</li> <li>– 0x00000080 - SVDIR</li> <li>– 0x00000100 - SVSTEER</li> <li>– 0x00000200 - SADATA</li> <li>– 0x00000400 - RTI</li> <li>– 0x00000800 - ALM_CORR</li> <li>– 0x00001000 - FREQ_BIAS_EST</li> </ul> </li> </ul>
<i>pCellDataMask</i>	<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>

## 8.459.2 Field Documentation

8.459.2.1 uint32\_t\* pack\_pds\_ResetPDSData\_t::pCellDataMask

8.459.2.2 uint32\_t\* pack\_pds\_ResetPDSData\_t::pGPSDataMask

## 8.460 pack\_pds\_SetEventReportCallback\_t Struct Reference

### Data Fields

- uint8\_t [posDataNmea](#)
- uint8\_t [rptPosData](#)

### 8.460.1 Detailed Description

Structure contain the parameter for the automatic tracking configuration for the NMEA COM port.

#### Parameters

<i>posDataNmea</i>	<ul style="list-style-type: none"> <li>• Report new position data in NMEA format.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>
<i>rptPosData</i>	<ul style="list-style-type: none"> <li>• Report new position data in raw format.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> </ul>

### 8.460.2 Field Documentation

8.460.2.1 uint8\_t pack\_pds\_SetEventReportCallback\_t::posDataNmea

8.460.2.2 uint8\_t pack\_pds\_SetEventReportCallback\_t::rptPosData

## 8.461 pack\_pds\_SetPDSDefaults\_t Struct Reference

### Data Fields

- uint32\_t [operation](#)
- uint8\_t [timeout](#)
- uint32\_t [interval](#)
- uint32\_t [accuracy](#)

### 8.461.1 Detailed Description

Structure containing session configuration parameters.

## 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>

## 8.461.2 Field Documentation

8.461.2.1 uint32\_t pack\_pds\_SetPDSDefaults\_t::accuracy

8.461.2.2 uint32\_t pack\_pds\_SetPDSDefaults\_t::interval

8.461.2.3 uint32\_t pack\_pds\_SetPDSDefaults\_t::operation

8.461.2.4 uint8\_t pack\_pds\_SetPDSDefaults\_t::timeout

## 8.462 pack\_pds\_SetPDSSState\_t Struct Reference

## Data Fields

- uint32\_t [enable](#)

## 8.462.1 Detailed Description

Structure contain PDS state parameters.

## Parameters

<i>enable</i>	<ul style="list-style-type: none"> <li>• Desired PDS state <ul style="list-style-type: none"> <li>– Zero - disable</li> <li>– Non-Zero - enable</li> </ul> </li> </ul>
---------------	--

## 8.462.2 Field Documentation

8.462.2.1 uint32\_t pack\_pds\_SetPDSSState\_t::enable

## 8.463 pack\_pds\_SetPortAutomaticTracking\_t Struct Reference



## Data Fields

- uint32\_t [bAuto](#)

### 8.463.1 Detailed Description

Structure contain the parameter for the automatic tracking configuration for the NMEA COM port.

#### Parameters

<i>bAuto</i>	<ul style="list-style-type: none"> <li>• Enable automatic tracking for NMEA COM port             <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>
--------------	--

### 8.463.2 Field Documentation

8.463.2.1 uint32\_t pack\_pds\_SetPortAutomaticTracking\_t::bAuto

## 8.464 pack\_pds\_SetServiceAutomaticTracking\_t Struct Reference

## Data Fields

- uint32\_t [bAuto](#)

### 8.464.1 Detailed Description

Structure contain the parameter to Sets the automatic tracking state.

#### Parameters

<i>bAuto</i>	<ul style="list-style-type: none"> <li>• Automatic tracking session started for service             <ul style="list-style-type: none"> <li>– 0x00 - Disabled</li> <li>– 0x01 - Enabled</li> </ul> </li> </ul>
--------------	---

### 8.464.2 Field Documentation

8.464.2.1 uint32\_t pack\_pds\_SetServiceAutomaticTracking\_t::bAuto

## 8.465 pack\_pds\_SetXTRAAutomaticDownload\_t Struct Reference

## Data Fields

- uint32\_t [bEnabled](#)
- uint16\_t [interval](#)

### 8.465.1 Detailed Description

Structure contain the parameter to set the XTRA automatic database download configuration.

#### Parameters

<i>bEnabled</i>	<ul style="list-style-type: none"> <li>Automatic XTRA download status               <ul style="list-style-type: none"> <li>0 - Disabled</li> <li>1 - Enabled</li> </ul> </li> </ul>
<i>interval</i>	<ul style="list-style-type: none"> <li>Interval (hours) between XTRA downloads</li> </ul>

### 8.465.2 Field Documentation

8.465.2.1 `uint32_t pack_pds_SetXTRAAutomaticDownload_t::bEnabled`

8.465.2.2 `uint16_t pack_pds_SetXTRAAutomaticDownload_t::interval`

## 8.466 `pack_pds_SetXTRANetwork_t` Struct Reference

### Data Fields

- `uint32_t` [preference](#)

### 8.466.1 Detailed Description

Structure contain the parameter to Sets the XTRA WWAN network preference

#### Parameters

<i>preference</i>	<ul style="list-style-type: none"> <li>XTRA WWAN network preference               <ul style="list-style-type: none"> <li>0x00 - None (any available network)</li> <li>0x01 - Home-only, only when on home systems</li> <li>0x02 - Roam-only, only when on non-home systems</li> </ul> </li> </ul>
-------------------	---

### 8.466.2 Field Documentation

8.466.2.1 `uint32_t pack_pds_SetXTRANetwork_t::preference`

## 8.467 `pack_pds_SLQSGetAGPSConfig_t` Struct Reference

### Data Fields

- `uint8_t *` [pNetworkMode](#)

### 8.467.1 Detailed Description

Structure contain the parameter to PDS AGPS (MS-based) configuration pack.

#### Parameters

<i>pNetworkMode</i>	<ul style="list-style-type: none"> <li>• Network Mode of AGPS Server [optional - should be present in Multimode Systems] <ul style="list-style-type: none"> <li>– 0x00 - UMTS</li> <li>– 0x01 - CDMA</li> </ul> </li> </ul>
---------------------	---

### 8.467.2 Field Documentation

8.467.2.1 uint8\_t\* pack\_pds\_SLQSPDSInjectAGPSConfig\_t::pNetworkMode

## 8.468 pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t Struct Reference

### Data Fields

- uint64\_t [timeMsec](#)
- uint32\_t [timeUncMsec](#)
- uint8\_t [timeBase](#)
- uint8\_t [forceFlag](#)

### 8.468.1 Detailed Description

Structure contain the parameter to Injects a absolute time reference.

#### Parameters

<i>timeMsec</i>	<ul style="list-style-type: none"> <li>• Represents the number of milliseconds elapsed since either a GPS or UTC time base. If the time base is UTC, this value should NOT include leap seconds</li> </ul>
<i>timeUncMsec</i>	<ul style="list-style-type: none"> <li>• Time uncertainty in milliseconds</li> </ul>
<i>timeBase</i>	<ul style="list-style-type: none"> <li>• Time base <ul style="list-style-type: none"> <li>– 0x00 - GPS (midnight, Jan 6, 1980)</li> <li>– 0x01 - UTC (midnight, Jan 1, 1970)</li> </ul> </li> </ul>
<i>forceFlag</i>	<ul style="list-style-type: none"> <li>• Force acceptance of data</li> </ul>

### 8.468.2 Field Documentation

8.468.2.1 uint8\_t pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t::forceFlag

8.468.2.2 uint8\_t pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t::timeBase

8.468.2.3 uint64\_t pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t::timeMsec

8.468.2.4 uint32\_t pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t::timeUncMsec

## 8.469 pack\_pds\_SLQSPDSInjectPositionData\_t Struct Reference

### Data Fields

- uint64\_t \* [pTimeStamp](#)
- uint64\_t \* [pLatitude](#)
- uint64\_t \* [pLongitude](#)
- uint32\_t \* [pAltitudeWrtEllipsoid](#)
- uint32\_t \* [pAltitudeWrtSealevel](#)
- uint32\_t \* [pHorizontalUncCircular](#)
- uint32\_t \* [pVerticalUnc](#)
- uint8\_t \* [pHorizontalConfidence](#)
- uint8\_t \* [pVerticalConfidence](#)
- uint8\_t \* [pPositionSource](#)
- uint8\_t \* [pTimeType](#)

### 8.469.1 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>

## 8.469.2 Field Documentation

8.469.2.1 uint32\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pAltitudeWrtEllipsoid

8.469.2.2 uint32\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pAltitudeWrtSealevel

8.469.2.3 uint8\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pHorizontalConfidence

8.469.2.4 uint32\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pHorizontalUncCircular

8.469.2.5 uint64\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pLatitude

8.469.2.6 uint64\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pLongitude

8.469.2.7 uint8\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pPositionSource

8.469.2.8 uint64\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pTimeStamp

8.469.2.9 uint8\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pTimeType

8.469.2.10 uint8\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pVerticalConfidence

8.469.2.11 uint32\_t\* pack\_pds\_SLQSPDSInjectPositionData\_t::pVerticalUnc

## 8.470 pack\_pds\_SLQSSetAGPSConfig\_t Struct Reference

### Data Fields

- uint32\_t \* [pServerAddress](#)
- uint32\_t \* [pServerPort](#)
- uint8\_t \* [pServerURL](#)
- uint8\_t \* [pServerURLLength](#)
- uint8\_t \* [pNetworkMode](#)

### 8.470.1 Detailed Description

Structure contain the parameter to Sets the PDS AGPS (MS-based) configuration.

#### Parameters

<i>pServerAddress</i>	<ul style="list-style-type: none"> <li>• IPv4 address of AGPS server [optional]</li> </ul>
<i>pServerPort</i>	<ul style="list-style-type: none"> <li>• Port number of AGPS server [optional - should be present when pServerAddress is present]</li> </ul>
<i>pServerURL</i>	<ul style="list-style-type: none"> <li>• URL of the AGPS server [optional]</li> </ul>
<i>pServerURL- Length</i>	<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>	<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>

### 8.470.2 Field Documentation

8.470.2.1 uint8\_t\* pack\_pds\_SLQSSetAGPSConfig\_t::pNetworkMode

8.470.2.2 uint32\_t\* pack\_pds\_SLQSSetAGPSConfig\_t::pServerAddress

8.470.2.3 uint32\_t\* pack\_pds\_SLQSSetAGPSConfig\_t::pServerPort

8.470.2.4 uint8\_t\* pack\_pds\_SLQSSetAGPSConfig\_t::pServerURL

8.470.2.5 uint8\_t\* pack\_pds\_SLQSSetAGPSConfig\_t::pServerURLLength

## 8.471 pack\_pds\_SLQSSetPositionMethodState\_t Struct Reference

### Data Fields

- uint8\_t \* [pXtraTimeState](#)

- uint8\_t \* [pXtraDataState](#)
- uint8\_t \* [pWifiState](#)

### 8.471.1 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>

### 8.471.2 Field Documentation

8.471.2.1 uint8\_t\* [pack\\_pds\\_SLQSSetPositionMethodState\\_t::pWifiState](#)

8.471.2.2 uint8\_t\* [pack\\_pds\\_SLQSSetPositionMethodState\\_t::pXtraDataState](#)

8.471.2.3 uint8\_t\* [pack\\_pds\\_SLQSSetPositionMethodState\\_t::pXtraTimeState](#)

## 8.472 pack\_pds\_StartPDSTrackingSessionExt\_t Struct Reference

### Data Fields

- uint8\_t [sessionControl](#)
- uint8\_t [sessionType](#)
- uint8\_t [sessionOperation](#)
- uint8\_t [sessionServerOption](#)
- uint8\_t [fixTimeout](#)
- uint32\_t [fixInterval](#)
- uint32\_t [fixCount](#)
- uint32\_t [fixAccuracy](#)

### 8.472.1 Detailed Description

Structure contain the parameter for PDS tracking session.

## Parameters

<i>sessionControl</i>	<ul style="list-style-type: none"> <li>Control method: <ul style="list-style-type: none"> <li>0x0 - Manual</li> </ul> </li> </ul>
<i>sessionType</i>	<ul style="list-style-type: none"> <li>Type: <ul style="list-style-type: none"> <li>0x0 - New</li> </ul> </li> </ul>
<i>session-Operation</i>	<ul style="list-style-type: none"> <li>Operating mode: <ul style="list-style-type: none"> <li>0x00 - Standalone</li> <li>0x01 - MS-based</li> </ul> </li> </ul>
<i>sessionServer-Option</i>	<ul style="list-style-type: none"> <li>Location server option: <ul style="list-style-type: none"> <li>0x0 - Default</li> </ul> </li> </ul>
<i>fixTimeout</i>	<ul style="list-style-type: none"> <li>Maximum time to work on each fix (in seconds, max 255)</li> </ul>
<i>fixCount</i>	<ul style="list-style-type: none"> <li>Count of position fix requests for this session (must be at least 1)</li> </ul>
<i>fixInterval</i>	<ul style="list-style-type: none"> <li>interval between position fix requests (in seconds)</li> </ul>
<i>fixAccuracy</i>	<ul style="list-style-type: none"> <li>Preferred accuracy threshold(in meters)</li> </ul>

## 8.472.2 Field Documentation

8.472.2.1 uint32\_t pack\_pds\_StartPDSTrackingSessionExt\_t::fixAccuracy

8.472.2.2 uint32\_t pack\_pds\_StartPDSTrackingSessionExt\_t::fixCount

8.472.2.3 uint32\_t pack\_pds\_StartPDSTrackingSessionExt\_t::fixInterval

8.472.2.4 uint8\_t pack\_pds\_StartPDSTrackingSessionExt\_t::fixTimeout

8.472.2.5 uint8\_t pack\_pds\_StartPDSTrackingSessionExt\_t::sessionControl

8.472.2.6 uint8\_t pack\_pds\_StartPDSTrackingSessionExt\_t::sessionOperation

8.472.2.7 uint8\_t pack\_pds\_StartPDSTrackingSessionExt\_t::sessionServerOption

8.472.2.8 uint8\_t pack\_pds\_StartPDSTrackingSessionExt\_t::sessionType

## 8.473 pack\_qmi\_t Struct Reference



## Data Fields

- uint16\_t [xid](#)
- int [timeout](#)
- uint16\_t [msgid](#)
- uint8\_t [svc](#)

### 8.473.1 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

### 8.473.2 Field Documentation

8.473.2.1 uint16\_t pack\_qmi\_t::msgid

8.473.2.2 uint8\_t pack\_qmi\_t::svc

8.473.2.3 int pack\_qmi\_t::timeout

8.473.2.4 uint16\_t pack\_qmi\_t::xid

## 8.474 pack\_qos\_BindDataPort\_t Struct Reference

## Data Fields

- qos\_BindDataPortPeripheralEndPointID\_t \* [pPeripheralEndPointID](#)
- qos\_BindDataPortMuxID\_t \* [pMuxID](#)
- qos\_BindDataPortSIODDataPort\_t \* [pSIODDataPort](#)

### 8.474.1 Detailed Description

Structure that contains the request Binds a control point to a data port.

#### Parameters

<i>pPeripheralEnd-PointID</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qos_BindDataPortPeripheralEndPointID_t</a> for more information.</li> </ul>
<i>pMuxID</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qos_BindDataPortMuxID_t</a> for more information.</li> </ul>
<i>pSIODDataPort</i>	<ul style="list-style-type: none"> <li>• See <a href="#">qos_BindDataPortSIODDataPort_t</a> for more information.</li> </ul>

### 8.474.2 Field Documentation

8.474.2.1 qos\_BindDataPortMuxID\_t\* pack\_qos\_BindDataPort\_t::pMuxID

8.474.2.2 qos\_BindDataPortPeripheralEndPointID\_t\* pack\_qos\_BindDataPort\_t::pPeripheralEndPointID

8.474.2.3 qos\_BindDataPortSIODataPort\_t\* pack\_qos\_BindDataPort\_t::pSIODataPort

## 8.475 pack\_qos\_SLQSQosSwiReadApnExtraParams\_t Struct Reference

### Data Fields

- uint32\_t [apnId](#)

### 8.475.1 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>
-------------------	--

### 8.475.2 Field Documentation

8.475.2.1 uint32\_t pack\_qos\_SLQSQosSwiReadApnExtraParams\_t::apnId

## 8.476 pack\_qos\_SLQSQosSwiReadDataStats\_t Struct Reference

### Data Fields

- uint32\_t [apnId](#)

### 8.476.1 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>
-------------------	--

### 8.476.2 Field Documentation

8.476.2.1 uint32\_t pack\_qos\_SLQSQosSwiReadDataStats\_t::apnId

## 8.477 pack\_qos\_SLQSSetQosEventCallback\_t Struct Reference

## Data Fields

- uint8\_t [enable](#)

## 8.477.1 Detailed Description

Structure that contains the APN ID to obtain data statistics

## Parameters

<i>in</i>	<i>enable</i>	<ul style="list-style-type: none"> <li>• 1 - Enable QoS event reporting</li> <li>• 0 - Disable QoS event reporting</li> </ul>
-----------	---------------	---

## 8.477.2 Field Documentation

8.477.2.1 uint8\_t pack\_qos\_SLQSSetQosEventCallback\_t::enable

## 8.478 pack\_rms\_SetSMSWake\_t Struct Reference

## Data Fields

- uint32\_t [enabled](#)
- uint32\_t [wake\\_mask](#)

## 8.478.1 Detailed Description

Pack set SMS wake parameters

## Parameters

<i>enabled</i>	<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>wake_mask</i>	<ul style="list-style-type: none"> <li>• SMS wake mask to search for incoming messages (only relevant when enabled)</li> </ul>

## 8.478.2 Field Documentation

8.478.2.1 uint32\_t pack\_rms\_SetSMSWake\_t::enabled

8.478.2.2 uint32\_t pack\_rms\_SetSMSWake\_t::wake\_mask

## 8.479 pack\_sar\_SLQSSetRfSarState\_t Struct Reference

## Data Fields

- uint32\_t [RfSarState](#)

### 8.479.1 Detailed Description

This sturcture contain sets the specified RF SAR state parameter.

#### Parameters

<i>RfSarState</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>
-------------------	--

### 8.479.2 Field Documentation

8.479.2.1 `uint32_t pack_sar_SLQSSetRfSarState_t::RfSarState`

## 8.480 `pack_sms_SaveSMS_t` Struct Reference

#### Data Fields

- `uint32_t` [storageType](#)
- `uint32_t` [messageFormat](#)
- `uint32_t` [messageSize](#)
- `uint8_t *` [pMessage](#)

### 8.480.1 Detailed Description

Sutstructure containing parameters to Save an SMS message to device memory

#### Parameters

<i>storageType</i>	<ul style="list-style-type: none"> <li>• SMS message storage type <ul style="list-style-type: none"> <li>– 0 - UIM - Invalid in case of CDMA device that does not require SIM</li> <li>– 1 - NV</li> </ul> </li> </ul>
<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>

<i>pMessage</i>	<ul style="list-style-type: none"> <li>• The message contents</li> </ul>
-----------------	--

## 8.480.2 Field Documentation

8.480.2.1 uint32\_t pack\_sms\_SaveSMS\_t::messageFormat

8.480.2.2 uint32\_t pack\_sms\_SaveSMS\_t::messageSize

8.480.2.3 uint8\_t\* pack\_sms\_SaveSMS\_t::pMessage

8.480.2.4 uint32\_t pack\_sms\_SaveSMS\_t::storageType

## 8.481 pack\_sms\_SendSMS\_t Struct Reference

### Data Fields

- uint32\_t [messageFormat](#)
- uint32\_t [messageSize](#)
- uint8\_t \* [pMessage](#)
- uint8\_t \* [pLinktimer](#)

### 8.481.1 Detailed Description

This structure contains pack send SMS parameters.

#### Parameters

<i>messageFormat</i>	<ul style="list-style-type: none"> <li>• Message format <ul style="list-style-type: none"> <li>– 0 - CDMA (IS-637B)</li> <li>– 1 - 5 (Reserved)</li> <li>– 6 - GSM/WCDMA PP</li> </ul> </li> </ul>
<i>messageSize</i>	<ul style="list-style-type: none"> <li>• The length of the message contents in bytes</li> </ul>
<i>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>

## 8.481.2 Field Documentation

8.481.2.1 uint32\_t pack\_sms\_SendSMS\_t::messageFormat

8.481.2.2 uint32\_t pack\_sms\_SendSMS\_t::messageSize

8.481.2.3 uint8\_t\* pack\_sms\_SendSMS\_t::pLinktimer

8.481.2.4 uint8\_t\* pack\_sms\_SendSMS\_t::pMessage

## 8.482 pack\_sms\_SetNewSMSCallback\_t Struct Reference

### Data Fields

- enum [eqmiCbkJSetStatus status](#)

### 8.482.1 Detailed Description

This strucure contains pack set new SMS callback parameters.

#### Parameters

<i>status</i>	callback parameter
---------------	--------------------

### 8.482.2 Field Documentation

8.482.2.1 enum [eqmiCbkJSetStatus](#) pack\_sms\_SetNewSMSCallback\_t::status

## 8.483 pack\_sms\_SetSMSCAddress\_t Struct Reference

### Data Fields

- uint8\_t \* [pSMSCAddress](#)
- uint8\_t \* [pSMSCType](#)

### 8.483.1 Detailed Description

Structure to Set the SMS center address.

#### Parameters

<i>SMSCAddress</i>	<ul style="list-style-type: none"> <li>The SMS center address represented as a NULL terminated string</li> </ul>
<i>SMSCType</i>	<ul style="list-style-type: none"> <li>The SMS center address type represented as a NULL terminated string (optional).</li> </ul>

### 8.483.2 Field Documentation

8.483.2.1 uint8\_t\* pack\_sms\_SetSMSCAddress\_t::pSMSCAddress

8.483.2.2 uint8\_t\* pack\_sms\_SetSMSCAddress\_t::pSMSCType

## 8.484 pack\_sms\_SLQSDelateSMS\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pMessageIndex](#)
- uint32\_t \* [pMessageTag](#)
- uint8\_t \* [pMessageMode](#)

## 8.484.1 Detailed Description

This structure contains pack delete SMS 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>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>

## 8.484.2 Field Documentation

8.484.2.1 uint32\_t\* pack\_sms\_SLQSDelateSMS\_t::pMessageIndex

8.484.2.2 uint8\_t\* pack\_sms\_SLQSDelateSMS\_t::pMessageMode

8.484.2.3 uint32\_t\* pack\_sms\_SLQSDelateSMS\_t::pMessageTag

8.484.2.4 uint32\_t pack\_sms\_SLQSDelateSMS\_t::storageType

## 8.485 pack\_sms\_SLQSGetSMS\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)
- uint8\_t \* [pMessageMode](#)

### 8.485.1 Detailed Description

This sturcture contains pack get SMS 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>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>

### 8.485.2 Field Documentation

8.485.2.1 uint32\_t pack\_sms\_SLQSGetSMS\_t::messageIndex

8.485.2.2 uint8\_t\* pack\_sms\_SLQSGetSMS\_t::pMessageMode

8.485.2.3 uint32\_t pack\_sms\_SLQSGetSMS\_t::storageType

## 8.486 pack\_sms\_SLQSGetSmsBroadcastConfig\_t Struct Reference

### Data Fields

- uint8\_t [mode](#)

### 8.486.1 Detailed Description

This structure contains get SMS boardcast configure parameter.

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>Mode <ul style="list-style-type: none"> <li>0x00 - CDMA, LTE (if network type is CDMA)</li> <li>0x01 - GW, LTE (if network type is UMTS)</li> </ul> </li> </ul>
-------------	--

### 8.486.2 Field Documentation

8.486.2.1 uint8\_t pack\_sms\_SLQSGetSmsBroadcastConfig\_t::mode

## 8.487 pack\_sms\_SLQSGetSMSList\_t Struct Reference



## Data Fields

- uint32\_t [storageType](#)
- uint32\_t \* [pRequestedTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.487.1 Detailed Description

This structure contains pack get SMS list 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>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>

### 8.487.2 Field Documentation

8.487.2.1 uint8\_t\* pack\_sms\_SLQSGetSMSList\_t::pMessageMode

8.487.2.2 uint32\_t\* pack\_sms\_SLQSGetSMSList\_t::pRequestedTag

8.487.2.3 uint32\_t pack\_sms\_SLQSGetSMSList\_t::storageType

## 8.488 pack\_sms\_SLQSMModifySMSStatus\_t Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)
- uint32\_t [messageTag](#)
- uint8\_t \* [pMessageMode](#)

### 8.488.1 Detailed Description

This structure contains pack modify SMS status 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>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>

## 8.488.2 Field Documentation

8.488.2.1 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageIndex

8.488.2.2 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::messageTag

8.488.2.3 uint8\_t\* pack\_sms\_SLQSMModifySMSStatus\_t::pMessageMode

8.488.2.4 uint32\_t pack\_sms\_SLQSMModifySMSStatus\_t::storageType

## 8.489 pack\_sms\_SLQSSendAsyncSMS\_t Struct Reference

## Data Fields

- [sms\\_sendAsynsmsParams](#) \* [pSendSmsParams](#)

## 8.489.1 Detailed Description

Structure contain Parameter to Send 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="#">sms_sendasynsmsparams</a></li> </ul>
------------------------	--

## 8.489.2 Field Documentation

8.489.2.1 [sms\\_sendAsynsmsParams](#)\* pack\_sms\_SLQSSendAsyncSMS\_t::pSendSmsParams

## 8.490 pack\_sms\_SLQSSetIndicationRegister\_t Struct Reference

## Data Fields

- [sms\\_setIndicationReg](#) \* [pSetIndicationRegReq](#)

## 8.490.1 Detailed Description

Structure contain Parameter that sets the registration state of different WMS indications.

## Parameters

<i>pSetIndication- Reg</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of indicationRegReqParams             <ul style="list-style-type: none"> <li>– See <a href="#">sms_setIndicationReg</a> for more information</li> </ul> </li> </ul>
--------------------------------	---

## 8.490.2 Field Documentation

8.490.2.1 [sms\\_setIndicationReg](#)\* [pack\\_sms\\_SLQSSetIndicationRegister\\_t::pSetIndicationRegReq](#)

## 8.491 pack\_sms\_SLQSSetSmsBroadcastActivation\_t Struct Reference

## Data Fields

- [uint8\\_t mode](#)
- [uint8\\_t broadcastActivate](#)

## 8.491.1 Detailed Description

Structure contain parameters that Enables or disables the reception of broadcast SMS messages.

## Parameters

<i>Mode</i>	<ul style="list-style-type: none"> <li>• Mode</li> <li>• 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>• 0x01 - GW, LTE (if network type is UMTS)</li> </ul>
<i>broadcast- Activate</i>	<ul style="list-style-type: none"> <li>• 0x00 - Disable broadcast</li> <li>• 0x01 - Activate broadcast</li> </ul>

## 8.491.2 Field Documentation

8.491.2.1 [uint8\\_t pack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t::broadcastActivate](#)

8.491.2.2 [uint8\\_t pack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t::mode](#)

## 8.492 pack\_sms\_SLQSSetSmsBroadcastConfig\_t Struct Reference

## Data Fields

- `uint8_t mode`
- `sms_qaQmi3GPPBroadcastCfgInfo * pBroadcastConfig`
- `sms_qaQmi3GPP2BroadcastCfgInfo * pCDMABroadcastConfig`

### 8.492.1 Detailed Description

Structure containing parameters that provides the information about the SMS BroadcastConfiguration

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"> <li>• Mode <ul style="list-style-type: none"> <li>– 0x00 - CDMA, LTE (if network type is CDMA)</li> <li>– 0x01 - GW, LTE (if network type is UMTS)</li> </ul> </li> </ul>
<i>pBroadcast-Config</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP Broadcast Information(Optional).</li> </ul>
<i>pCDMA-BroadcastConfig</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP2 Broadcast Information(Optional).</li> </ul>

### 8.492.2 Field Documentation

8.492.2.1 `uint8_t pack_sms_SLQSSetSmsBroadcastConfig_t::mode`

8.492.2.2 `sms_qaQmi3GPPBroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pBroadcastConfig`

8.492.2.3 `sms_qaQmi3GPP2BroadcastCfgInfo* pack_sms_SLQSSetSmsBroadcastConfig_t::pCDMABroadcastConfig`

## 8.493 pack\_sms\_SLQSSetSmsStorage\_t Struct Reference

### Data Fields

- `uint8_t smsStorage`

### 8.493.1 Detailed Description

Structure contain Parameter to set the SMS Storage on the device

#### Parameters

<i>smsStorage</i>	<ul style="list-style-type: none"> <li>• SMS Storage <ul style="list-style-type: none"> <li>– 0x01 - device's permanent memory</li> <li>– 0x02 - UICC</li> </ul> </li> </ul>
-------------------	--

### 8.493.2 Field Documentation

8.493.2.1 uint8\_t pack\_sms\_SLQSSetSmsStorage\_t::smsStorage

## 8.494 pack\_sms\_SLQSSmsGetMaxStorageSize\_t Struct Reference

### Data Fields

- [sms\\_maxStorageSizeReq](#) \* [pMaxStorageSizeReq](#)

### 8.494.1 Detailed Description

This structure contains get maximum storage size.

#### Parameters

<i>pMaxStorage-SizeReq</i>	<ul style="list-style-type: none"> <li>• Request parameters for SmsSLQSGetMaxStorageSize             <ul style="list-style-type: none"> <li>– See <a href="#">sms_maxStorageSizeReq</a> for more information</li> </ul> </li> </ul>
----------------------------	---

### 8.494.2 Field Documentation

8.494.2.1 [sms\\_maxStorageSizeReq](#)\* [pack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t::pMaxStorageSizeReq](#)

## 8.495 pack\_sms\_SLQSSmsSetRoutes\_t Struct Reference

### Data Fields

- [sms\\_setRoutesReq](#) \* [pSetRoutesReq](#)

### 8.495.1 Detailed Description

Structure contain parameter that 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>	<ul style="list-style-type: none"> <li>• Pointer to structure of smsSetRoutesReq             <ul style="list-style-type: none"> <li>– See <a href="#">sms_setRoutesReq</a> for more information</li> </ul> </li> </ul>
----------------------	--

### 8.495.2 Field Documentation

8.495.2.1 [sms\\_setRoutesReq](#)\* [pack\\_sms\\_SLQSSmsSetRoutes\\_t::pSetRoutesReq](#)

## 8.496 pack\_swiaudio\_SLQSGetM2MAudioProfile\_t Struct Reference

### Data Fields

- uint8\_t \* [pGenerator](#)

### 8.496.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile pack 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>
------------------------------	--

### 8.496.2 Field Documentation

8.496.2.1 `uint8_t* pack_swiaudio_SLQSGetM2MAudioProfile_t::pGenerator`

## 8.497 `pack_swiaudio_SLQSGetM2MAudioVolume_t` Struct Reference

#### Data Fields

- `uint8_t` [Profile](#)
- `uint8_t` [Generator](#)

### 8.497.1 Detailed Description

This structure contains the SLQSGetM2MAudioVolume pack 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>

### 8.497.2 Field Documentation

8.497.2.1 `uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Generator`

8.497.2.2 `uint8_t pack_swiaudio_SLQSGetM2MAudioVolume_t::Profile`

## 8.498 `pack_swiaudio_SLQSGetM2MAVMute_t` Struct Reference

#### Data Fields

- `uint8_t` [Profile](#)

### 8.498.1 Detailed Description

This structure contains the SLQSGetM2MAVMute pack 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>
----------------	---

## 8.498.2 Field Documentation

8.498.2.1 uint8\_t pack\_swiaudio\_SLQSGetM2MAVMute\_t::Profile

## 8.499 pack\_swiaudio\_SLQSGetM2MSpkrGain\_t Struct Reference

## Data Fields

- uint8\_t [Profile](#)

## 8.499.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain pack 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>
----------------	---

## 8.499.2 Field Documentation

8.499.2.1 uint8\_t pack\_swiaudio\_SLQSGetM2MSpkrGain\_t::Profile

## 8.500 pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t Struct Reference

## Data Fields

- uint8\_t [Profile](#)
- uint8\_t [Device](#)
- uint8\_t [PIFACEId](#)
- [swiaudio\\_PCMparams](#) \* [pPCMPParams](#)

## 8.500.1 Detailed Description

This structure contains the SLQSSetM2MAudioAVCFG pack 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="#">swiaudio_PCMparams</a> for more information</li> </ul>

## 8.500.2 Field Documentation

8.500.2.1 `uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Device`

8.500.2.2 `uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::PIFACEId`

8.500.2.3 `swiaudio_PCMparams* pack_swiaudio_SLQSSetM2MAudioAVCFG_t::pPCMPParams`

8.500.2.4 `uint8_t pack_swiaudio_SLQSSetM2MAudioAVCFG_t::Profile`

## 8.501 `pack_swiaudio_SLQSSetM2MAudioLPBK_t` Struct Reference

### Data Fields

- `uint8_t Enable`

### 8.501.1 Detailed Description

This structure contains the SLQSSetM2MAudioLPBK pack 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>
---------------	--

### 8.501.2 Field Documentation

8.501.2.1 `uint8_t pack_swiaudio_SLQSSetM2MAudioLPBK_t::Enable`

## 8.502 `pack_swiaudio_SLQSSetM2MAudioProfile_t` Struct Reference

### Data Fields

- `uint8_t Profile`



- uint8\_t \* [pEarMute](#)
- uint8\_t \* [pMicMute](#)
- uint8\_t \* [pGenerator](#)
- uint8\_t \* [pVolume](#)
- uint8\_t \* [pCwtMute](#)

### 8.502.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack 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>

### 8.502.2 Field Documentation

8.502.2.1 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::pCwtMute

8.502.2.2 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::pEarMute

8.502.2.3 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::pGenerator

8.502.2.4 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::pMicMute

8.502.2.5 uint8\_t pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::Profile

8.502.2.6 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAudioProfile\_t::pVolume

## 8.503 pack\_swiaudio\_SLQSSetM2MAudioVolume\_t Struct Reference

### Data Fields

- uint8\_t [Profile](#)
- uint8\_t [Generator](#)
- uint8\_t [Level](#)

### 8.503.1 Detailed Description

This structure contains the SLQSSetM2MAudioProfile pack 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>

### 8.503.2 Field Documentation

8.503.2.1 uint8\_t pack\_swiaudio\_SLQSSetM2MAudioVolume\_t::Generator

8.503.2.2 uint8\_t pack\_swiaudio\_SLQSSetM2MAudioVolume\_t::Level

8.503.2.3 uint8\_t pack\_swiaudio\_SLQSSetM2MAudioVolume\_t::Profile

## 8.504 pack\_swiaudio\_SLQSSetM2MAVMute\_t Struct Reference

### Data Fields

- uint8\_t [Profile](#)
- uint8\_t [EarMute](#)
- uint8\_t [MicMute](#)
- uint8\_t \* [pCwtMute](#)

### 8.504.1 Detailed Description

This structure contains the SLQSSetM2MAVMute pack 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>

### 8.504.2 Field Documentation

8.504.2.1 uint8\_t pack\_swiaudio\_SLQSSetM2MAVMute\_t::EarMute

8.504.2.2 uint8\_t pack\_swiaudio\_SLQSSetM2MAVMute\_t::MicMute

8.504.2.3 uint8\_t\* pack\_swiaudio\_SLQSSetM2MAVMute\_t::pCwtMute

8.504.2.4 uint8\_t pack\_swiaudio\_SLQSSetM2MAVMute\_t::Profile

## 8.505 pack\_swiaudio\_SLQSSetM2MSpkrGain\_t Struct Reference

### Data Fields

- uint8\_t [Profile](#)
- uint16\_t [Value](#)

### 8.505.1 Detailed Description

This structure contains the SLQSSetM2MSpkrGain pack 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>

### 8.505.2 Field Documentation

8.505.2.1 uint8\_t pack\_swiaudio\_SLQSSetM2MSpkrGain\_t::Profile

8.505.2.2 uint16\_t pack\_swiaudio\_SLQSSetM2MSpkrGain\_t::Value

## 8.506 pack\_swiaavms\_SLQSAVMSSendSelection\_t Struct Reference

### Data Fields

- uint8\_t [selection](#)
- uint32\_t \* [pDeferTime](#)
- uint8\_t \* [pClientPerformOperationFlag](#)
- uint8\_t \* [pPackageID](#)
- uint16\_t \* [pRejectReason](#)

### 8.506.1 Detailed Description

Structure containing the AVMS selection

#### Parameters

<i>selection</i> [IN]	<ul style="list-style-type: none"> <li>• User 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/-OPTIONAL]	<ul style="list-style-type: none"> <li>• Defer time in minutes. A value of 0 will cause the prompt to be resent immediately.</li> </ul>
<i>pClientPerformOperationFlag</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> <li>• Client operation flag after accept. <ul style="list-style-type: none"> <li>– 0: if modem performs the operation (download or update)</li> <li>– 1: if client performs the operation (download or update)</li> </ul> </li> </ul>
<i>pPackageID</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> <li>• Package ID.</li> </ul>
<i>pRejectReason</i> [IN/-OPTIONAL]	<ul style="list-style-type: none"> <li>• Reject Reason.</li> </ul>

### 8.506.2 Field Documentation

8.506.2.1 uint8\_t\* pack\_swiaavms\_SLQSAVMSSendSelection\_t::pClientPerformOperationFlag

8.506.2.2 uint32\_t\* pack\_swiaavms\_SLQSAVMSSendSelection\_t::pDeferTime

8.506.2.3 uint8\_t\* pack\_swiaavms\_SLQSAVMSSendSelection\_t::pPackageID

8.506.2.4 uint16\_t\* pack\_swiaavms\_SLQSAVMSSendSelection\_t::pRejectReason

8.506.2.5 uint8\_t pack\_swiaavms\_SLQSAVMSSendSelection\_t::selection

## 8.507 pack\_swiaavms\_SLQSAVMSSetSettings\_t Struct Reference

## Data Fields

- uint8\_t [AutoConnect](#)
- uint8\_t [AutoReboot](#)
- uint8\_t [PromptFwDownload](#)
- uint8\_t [PromptFwUpdate](#)
- uint8\_t \* [pFwAutoSDM](#)
- uint32\_t \* [pPollingTimer](#)
- [PackSwiAvmsSetSettingsConnectionRetryTimers](#) \* [pConnectionRetryTimers](#)
- [PackSwiAvmsSetSettingsAPNInfo](#) \* [pAPNInfo](#)
- uint8\_t \* [pNotifStore](#)
- [PackSwiAvmsSetSettingsPeriodInfo](#) \* [pPeriodInfo](#)

## 8.507.1 Detailed Description

Structure containing the AVMS settings to be set on the device This maps to structure SLQSAVMSSetSettings

## Parameters

<i>AutoConnect</i> [IN]	<ul style="list-style-type: none"> <li>• Auto Connect to AirVantage server <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>AutoReboot</i> [IN]	<ul style="list-style-type: none"> <li>• Automatic device reboot when the request is received from AirVantage server <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>PromptFw-Download</i> [IN]	<ul style="list-style-type: none"> <li>• Firmware Auto Download <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>PromptFw-Update</i> [IN]	<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>pFwAutoSDM</i> [I-N/OPTIONAL]	<ul style="list-style-type: none"> <li>• 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>pPollingTimer</i> [I-N/OPTIONAL]	<ul style="list-style-type: none"> <li>• Polling timer to connect to AVMS server <ul style="list-style-type: none"> <li>– 0-525600 (min)</li> <li>– 0:disabled</li> </ul> </li> </ul>
<i>pConnection-RetryTimers</i> [IN-OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsConnectionRetryTimers</a> for more information</li> </ul>

<i>pAPNInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>
<i>pNotifStore</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>
<i>pPeriodInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>

## 8.507.2 Field Documentation

8.507.2.1 `uint8_t pack_swiavms_SLQSAVMSSetSettings_t::AutoConnect`

8.507.2.2 `uint8_t pack_swiavms_SLQSAVMSSetSettings_t::AutoReboot`

8.507.2.3 `PackSwiAvmsSetSettingsAPNInfo* pack_swiavms_SLQSAVMSSetSettings_t::pAPNInfo`

8.507.2.4 `PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiavms_SLQSAVMSSetSettings_t::pConnection-RetryTimers`

8.507.2.5 `uint8_t* pack_swiavms_SLQSAVMSSetSettings_t::pFwAutoSDM`

8.507.2.6 `uint8_t* pack_swiavms_SLQSAVMSSetSettings_t::pNotifStore`

8.507.2.7 `PackSwiAvmsSetSettingsPeriodInfo* pack_swiavms_SLQSAVMSSetSettings_t::pPeriodInfo`

8.507.2.8 `uint32_t* pack_swiavms_SLQSAVMSSetSettings_t::pPollingTimer`

8.507.2.9 `uint8_t pack_swiavms_SLQSAVMSSetSettings_t::PromptFwDownload`

8.507.2.10 `uint8_t pack_swiavms_SLQSAVMSSetSettings_t::PromptFwUpdate`

## 8.508 `pack_swiavms_SLQSAVMSSetSettings_v2_t` Struct Reference

### Data Fields

- `uint8_t AutoConnect`
- `uint8_t PromptFwDownload`
- `uint8_t PromptFwUpdate`
- `uint8_t * pFwAutoSDM`
- `uint32_t * pPollingTimer`
- `PackSwiAvmsSetSettingsConnectionRetryTimers * pConnectionRetryTimers`
- `PackSwiAvmsSetSettingsAPNInfo * pAPNInfo`
- `uint8_t * pNotifStore`
- `PackSwiAvmsSetSettingsPeriodInfo * pPeriodInfo`
- `uint8_t * pAutoReboot`

### 8.508.1 Detailed Description

Structure containing the AVMS settings to be set on the device This maps to structure `SLQSAVMSSetSettings_v2` (For AVC2 service)

## Parameters

<i>AutoConnect</i> [IN]	<ul style="list-style-type: none"> <li>• Auto Connect to AirVantage server <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>PromptFw-Download</i> [IN]	<ul style="list-style-type: none"> <li>• Firmware Auto Download <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>
<i>PromptFw-Update</i> [IN]	<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>pFwAutoSDM</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• 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>pPollingTimer</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• Polling timer to connect to AVMS server <ul style="list-style-type: none"> <li>– 0-525600 (min)</li> <li>– 0:disabled</li> </ul> </li> </ul>
<i>pConnection-RetryTimers</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsConnectionRetryTimers</a> for more information</li> </ul>
<i>pAPNInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>
<i>pNotifStore</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>
<i>pPeriodInfo</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAvmsSetSettingsAPNInfo</a> for more information</li> </ul>
<i>pAutoReboot</i> [IN/OPTIONAL]	<ul style="list-style-type: none"> <li>• Automatic device reboot when the request is received from AirVantage server <ul style="list-style-type: none"> <li>– 0x00 - FALSE</li> <li>– 0x01 - TRUE</li> </ul> </li> </ul>

## Note

Setting PromptFwDownload/PromptFwUpdate as TRUE implies AutoConnect to be also TRUE even if AutoConnect was set FALSE initially. Automatic firmware download/update selection trumps the connection setting flag.

## 8.508.2 Field Documentation

- 8.508.2.1 `uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::AutoConnect`
- 8.508.2.2 `PackSwiAvmsSetSettingsAPNInfo* pack_swiavms_SLQSAVMSSetSettings_v2_t::pAPNInfo`
- 8.508.2.3 `uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pAutoReboot`
- 8.508.2.4 `PackSwiAvmsSetSettingsConnectionRetryTimers* pack_swiavms_SLQSAVMSSetSettings_v2_t::pConnectionRetryTimers`
- 8.508.2.5 `uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pFwAutoSDM`
- 8.508.2.6 `uint8_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pNotifStore`
- 8.508.2.7 `PackSwiAvmsSetSettingsPeriodInfo* pack_swiavms_SLQSAVMSSetSettings_v2_t::pPeriodInfo`
- 8.508.2.8 `uint32_t* pack_swiavms_SLQSAVMSSetSettings_v2_t::pPollingTimer`
- 8.508.2.9 `uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::PromptFwDownload`
- 8.508.2.10 `uint8_t pack_swiavms_SLQSAVMSSetSettings_v2_t::PromptFwUpdate`

## 8.509 `pack_swiavms_SLQSAVMSStartSession_t` Struct Reference

### Data Fields

- `uint8_t sessionType`

### 8.509.1 Detailed Description

Structure that contains the session type request for AVMS 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. This field is mandatory in OMA-DM case but not necessary in LWM2M one. To keep compatibility, this shall be used in LWM2M without any specific treatment.</li> </ul> </li> </ul>
------------------------	--

### 8.509.2 Field Documentation

- 8.509.2.1 `uint8_t pack_swiavms_SLQSAVMSStartSession_t::sessionType`

## 8.510 `pack_swiavms_SLQSAVMSStopSession_t` Struct Reference

### Data Fields

- `uint8_t sessionType`

### 8.510.1 Detailed Description

Structure that contains the session type for AVMS Stop session command



## Parameters

<i>sessionType</i>	<ul style="list-style-type: none"> <li>Session type <ul style="list-style-type: none"> <li>0x01 - FOTA, to suspend FOTA session.</li> <li>0xFF - Suspend ongoing FOTA session or stop any other active AVMS session.</li> </ul> </li> </ul>
<i>resultcode</i>	<ul style="list-style-type: none"> <li>Tlv Result Code</li> </ul>

## 8.510.2 Field Documentation

8.510.2.1 uint8\_t pack\_swiaavms\_SLQSAVMSStopSession\_t::sessionType

## 8.511 pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t Struct Reference

## Data Fields

- uint32\_t [timeout](#)
- uint32\_t [resetDelay](#)
- uint8\_t [enable](#)

## 8.511.1 Detailed Description

This structure contains the set hardware watchdog settings request parameter.

## Parameters

<i>timeout</i>	<ul style="list-style-type: none"> <li>timeout value for HW watchdog (unit in second)</li> </ul>
<i>resetDelay</i>	<ul style="list-style-type: none"> <li>delay before reset after watchdog timeout (unit in second)</li> </ul>
<i>enable</i>	<ul style="list-style-type: none"> <li>0 to disable watchdog; 1 to enable watchdog</li> </ul>

## 8.511.2 Field Documentation

8.511.2.1 uint8\_t pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t::enable

8.511.2.2 uint32\_t pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t::resetDelay

8.511.2.3 uint32\_t pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t::timeout

## 8.512 pack\_swidms\_SLQSSwiDmsSetMTU\_t Struct Reference

## Data Fields

- uint16\_t [MTUSize](#)

### 8.512.1 Detailed Description

This structure contains the Set MTU request parameter.

#### Parameters

<i>MTUSize</i>	<ul style="list-style-type: none"><li>• Set MTU size</li><li>• MTU Values<ul style="list-style-type: none"><li>– 0 - use default value</li><li>– 576 to 2000 - other values required by carrier.</li><li>– Set the same MTU for all RAT/interfaces.</li><li>– New MTU size effective after modem reboot.</li></ul></li></ul>
----------------	--

### 8.512.2 Field Documentation

8.512.2.1 `uint16_t pack_swidms_SLQSSwiDmsSetMTU_t::MTUSize`

## 8.513 `pack_swidms_SLQSSwiDmsSetUsbComp_t` Struct Reference

#### Data Fields

- `uint32_t CfgValue`

### 8.513.1 Detailed Description

This structure contains the Set Usb Interface Config request parameter.

#### Parameters

---

<i>CfgValue</i>	<ul style="list-style-type: none"> <li>• Host composition bit mask, see supported USB interface bitmasks, not supported by QMI object setting for 9x50 modules <ul style="list-style-type: none"> <li>– 0x00000001 - DIAG interface</li> <li>– 0x00000002 - ADB interface</li> <li>– 0x00000004 - NMEA interface</li> <li>– 0x00000008 - MODEM interface</li> <li>– 0x00000010 - RESERVED5</li> <li>– 0x00000020 - RESERVED6</li> <li>– 0x00000040 - RESERVED7</li> <li>– 0x00000080 - RESERVED8</li> <li>– 0x00000100 - RMENT0 interface</li> <li>– 0x00000200 - RESERVED10</li> <li>– 0x00000400 - RMENT1 interface</li> <li>– 0x00000800 - RESERVED12</li> <li>– 0x00001000 - MBIM interface</li> <li>– 0x00002000 - RESERVED14</li> <li>– 0x00004000 - RNDIS interface</li> <li>– 0x00008000 - RESERVED16</li> <li>– 0x00010000 - AUDIO interface</li> <li>– 0x00020000 - RESERVED18</li> <li>– 0x00080000 - ECM interface</li> <li>– 0x00100000 - RESERVED21</li> <li>– 0x00200000 - RESERVED22</li> <li>– 0xFFC00000 - RESERVED</li> </ul> </li> </ul>
-----------------	--

## 8.513.2 Field Documentation

8.513.2.1 uint32\_t pack\_swidms\_SLQSSwiDmsSetUsbComp\_t::CfgValue

## 8.514 pack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t Struct Reference

### Data Fields

- uint8\_t [nUsbNetNum](#)

### 8.514.1 Detailed Description

This structure contains usb net number to set on remote endpoint for QMAP configuration

#### Parameters

<i>nUsbNetNum</i>	<ul style="list-style-type: none"> <li>• value of usb net numbers to set for the device</li> <li>• range is 0-8</li> </ul>
-------------------	--

## 8.514.2 Field Documentation

8.514.2.1 `uint8_t pack_swidms_SLQSSwiDmsSetUsbNetNum_t::nUsbNetNum`

## 8.515 `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](#)

### 8.515.1 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>

## 8.515.2 Field Documentation

8.515.2.1 uint32\_t pack\_swiloc\_SwiLocSetAutoStart\_t::fix\_rate

8.515.2.2 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::fix\_type

8.515.2.3 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::function

8.515.2.4 uint32\_t pack\_swiloc\_SwiLocSetAutoStart\_t::max\_dist

8.515.2.5 uint8\_t pack\_swiloc\_SwiLocSetAutoStart\_t::max\_time

8.515.2.6 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_fix\_rate

8.515.2.7 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_fix\_type

8.515.2.8 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_function

8.515.2.9 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_max\_dist

8.515.2.10 int pack\_swiloc\_SwiLocSetAutoStart\_t::set\_max\_time

## 8.516 pack\_swima\_SLQSOMADMCancelSession\_t Struct Reference

### Data Fields

- uint32\_t [sessionType](#)

### 8.516.1 Detailed Description

Structure that contains the session type for OMA cancel session command

## 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>0xFF - Cancel any active OMADM session</li> </ul> </li> </ul>
--------------------	--

## 8.516.2 Field Documentation

8.516.2.1 uint32\_t pack\_swisma\_SLQSOMADMCancelSession\_t::sessionType

## 8.517 pack\_swisma\_SLQSOMADMCancelSessionExt\_t Struct Reference

## Data Fields

- uint32\_t [sessionType](#)

## 8.517.1 Detailed Description

Structure that contains the session type for OMA cancel session command

## 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>0xFF - Cancel any active OMADM session</li> </ul> </li> </ul>
--------------------	--

## 8.517.2 Field Documentation

8.517.2.1 uint32\_t pack\_swisma\_SLQSOMADMCancelSessionExt\_t::sessionType

## 8.518 pack\_swisma\_SLQSOMADMGetSessionInfo\_t Struct Reference

## Data Fields

- uint32\_t [SessionType](#)

## 8.518.1 Detailed Description

Structure that contains the session type for OMA get session info command

## 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 no active sessions are available, then previous OMADM session info is returned</li> </ul> </li> </ul>
--------------------	--

## 8.518.2 Field Documentation

8.518.2.1 uint32\_t pack\_swisma\_SLQSOMADMGetSessionInfo\_t::SessionType

## 8.519 pack\_swisma\_SLQSOMADMSelect\_t Struct Reference

### Data Fields

- uint32\_t [selection](#)
- uint32\_t \* [pDeferTime](#)
- uint32\_t \* [pRejectReason](#)

### 8.519.1 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>

## 8.519.2 Field Documentation

8.519.2.1 uint32\_t\* pack\_swisma\_SLQSOMADMSelect\_t::pDeferTime

8.519.2.2 uint32\_t\* pack\_swisma\_SLQSOMADMSelect\_t::pRejectReason

8.519.2.3 uint32\_t pack\_swisma\_SLQSOMADMSelect\_t::selection

## 8.520 pack\_swisma\_SLQSOMADMSelectExt\_t Struct Reference

### Data Fields

- uint32\_t [selection](#)

### 8.520.1 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>
------------------	---

## 8.520.2 Field Documentation

8.520.2.1 `uint32_t pack_swioama_SLQSOMADM_SendSelectionExt_t::selection`

8.521 `pack_swioama_SLQSOMADM_SetSettings_t` Struct Reference

## Data Fields

- `uint8_t FOTAdownload`
- `uint8_t FOTAupdate`
- `uint8_t * pAutosdm`
- `uint8_t * pFwAutoCheck`

## 8.521.1 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</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</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>



### 8.521.2 Field Documentation

8.521.2.1 uint8\_t pack\_swisma\_SLQSOMADMSetSettings\_t::FOTAdownload

8.521.2.2 uint8\_t pack\_swisma\_SLQSOMADMSetSettings\_t::FOTAUpdate

8.521.2.3 uint8\_t\* pack\_swisma\_SLQSOMADMSetSettings\_t::pAutosdm

8.521.2.4 uint8\_t\* pack\_swisma\_SLQSOMADMSetSettings\_t::pFwAutoCheck

## 8.522 pack\_swisma\_SLQSOMADMSetSettingsExt\_t Struct Reference

### Data Fields

- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t [OMADMEnable](#)
- uint8\_t [OMADMLogEnable](#)
- uint8\_t [FUMOEnable](#)
- uint8\_t [PRLEnable](#)

### 8.522.1 Detailed Description

Structure containing the OMA DM settings to be set on the device This maps to structure SLQSOMADMSetSettings-Ext

#### 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>OMADMEnable</i>	[IN] <ul style="list-style-type: none"> <li>• 1 byte parameter indicating OMA DM <ul style="list-style-type: none"> <li>– 0x00 - OMA DM Disabled</li> <li>– 0x01 - OMA DM Enabled</li> </ul> </li> </ul>
<i>OMADMLog-Enable</i>	[IN] <ul style="list-style-type: none"> <li>• 1 byte parameter indicating OMA DM Logs <ul style="list-style-type: none"> <li>– 0x00 - OMA DM Logs Disabled</li> <li>– 0x01 - OMA DM Logs Enabled</li> </ul> </li> </ul>
<i>FUMOEnable</i>	[IN] <ul style="list-style-type: none"> <li>• 1 byte parameter indicating FUMO enabled <ul style="list-style-type: none"> <li>– 0x00 - FUMO Disabled</li> <li>– 0x01 - FUMO enabled</li> </ul> </li> </ul>

<i>PRLenable</i>	[IN] <ul style="list-style-type: none"> <li>• 1 byte parameter indicating PRL Enabled             <ul style="list-style-type: none"> <li>– 0x00 - PRL Disabled</li> <li>– 0x01 - PRL Enabled</li> <li>– 0x02 - Launch a CI PRL</li> <li>– 0x03 - PRL Enabled and PRL automatic update every 45 days</li> <li>– 0x04 - PRL Enabled and PRL update ever 90 days</li> </ul> </li> </ul>
------------------	--

### 8.522.2 Field Documentation

8.522.2.1 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::FOTAdownload

8.522.2.2 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::FOTAUpdate

8.522.2.3 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::FUMOEnable

8.522.2.4 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::OMADMEEnable

8.522.2.5 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::OMADMLogEnable

8.522.2.6 uint8\_t pack\_swima\_SLQSOMADMSetSettingsExt\_t::PRLenable

## 8.523 pack\_swima\_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</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>
-------------------------	---

### 8.523.2 Field Documentation

8.523.2.1 uint32\_t pack\_swima\_SLQSOMADMStartSession\_t::sessionType

## 8.524 pack\_swima\_SLQSOMADMStartSessionExt\_t Struct Reference

### Data Fields

- uint32\_t [sessionType](#)

### 8.524.1 Detailed Description

Structure that contains the session type for OMA start session command

#### Parameters

<i>sessionType</i>	<div> <div>[IN]</div> <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> </div>
--------------------	---

### 8.524.2 Field Documentation

8.524.2.1 `uint32_t pack_swioa_SLQSOMADMStartSessionExt_t::sessionType`

## 8.525 pack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t Struct Reference

### Data Fields

- `uint8_t mitigationDevIDLen`
- `char mitigationDevID [255]`

### 8.525.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>mitigationDevID-Len</i>	<div> <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> </div>
<i>mitigationDevID</i>	<div> <ul style="list-style-type: none"><li>Mitigation device ID</li></ul> </div>

### 8.525.2 Field Documentation

8.525.2.1 `char pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevID[255]`

8.525.2.2 `uint8_t pack_tmd_SLQSTmdDeRegNotMitigationLvl_t::mitigationDevIDLen`

## 8.526 pack\_tmd\_SLQSTmdGetMitigationLvl\_t Struct Reference

### Data Fields

- `uint8_t mitigationDevIDLen`
- `char mitigationDevID [255]`

### 8.526.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### 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>

### 8.526.2 Field Documentation

8.526.2.1 char pack\_tmd\_SLQSTmdGetMitigationLvl\_t::mitigationDevID[255]

8.526.2.2 uint8\_t pack\_tmd\_SLQSTmdGetMitigationLvl\_t::mitigationDevIDLen

## 8.527 pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t Struct Reference

#### Data Fields

- uint8\_t [mitigationDevIDLen](#)
- char [mitigationDevID](#) [255]

### 8.527.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### 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>

### 8.527.2 Field Documentation

8.527.2.1 char pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t::mitigationDevID[255]

8.527.2.2 uint8\_t pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t::mitigationDevIDLen

## 8.528 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](#)

### 8.528.1 Detailed Description

This structure contains information of the request parameters associated with a Change PIN API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>changePIN</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_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> <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.

### 8.528.2 Field Documentation

8.528.2.1 [uim\\_changeUIMPIN](#) pack\_uim\_ChangePin\_t::changePIN

8.528.2.2 [uim\\_encryptedPIN1](#) pack\_uim\_ChangePin\_t::EncryptedPIN1

8.528.2.3 [uint32\\_t\\*](#) pack\_uim\_ChangePin\_t::pIndicationToken

8.528.2.4 [uint8\\_t\\*](#) pack\_uim\_ChangePin\_t::pKeyReferenceID

8.528.2.5 [uim\\_sessionInformation](#) pack\_uim\_ChangePin\_t::sessionInfo

8.528.2.6 [uint16\\_t](#) pack\_uim\_ChangePin\_t::Tlvresult

## 8.529 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](#)

### 8.529.1 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="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_fileInfo</a> for more information.</li> </ul>
<i>readTransparent</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_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> <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>
<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.

### 8.529.2 Field Documentation

8.529.2.1 `uim_fileInfo` `pack_uim_ReadTransparent_t::fileIndex`

8.529.2.2 `uint8_t*` `pack_uim_ReadTransparent_t::pEncryptData`

8.529.2.3 `uint32_t*` `pack_uim_ReadTransparent_t::pIndicationToken`

8.529.2.4 `uim_readTransparentInfo` `pack_uim_ReadTransparent_t::readTransparent`

8.529.2.5 `uim_sessionInformation` `pack_uim_ReadTransparent_t::sessionInfo`

8.529.2.6 `uint16_t` `pack_uim_ReadTransparent_t::Tlvresult`

## 8.530 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](#)

### 8.530.1 Detailed Description

This structure contains information of the request parameters associated with a set pin protection API.

#### Parameters

<i>sessionInfo</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_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> <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\\_SetPinProtection\\_t::EncryptedPIN1](#)

8.530.2.2 [uint32\\_t\\*](#) [pack\\_uim\\_SetPinProtection\\_t::pIndicationToken](#)

8.530.2.3 [uim\\_setPINProtection](#) [pack\\_uim\\_SetPinProtection\\_t::pinProtection](#)

8.530.2.4 [uint8\\_t\\*](#) [pack\\_uim\\_SetPinProtection\\_t::pKeyReferenceID](#)

8.530.2.5 [uim\\_sessionInformation](#) [pack\\_uim\\_SetPinProtection\\_t::sessionInfo](#)

8.530.2.6 `uint16_t pack_uim_SetPinProtection_t::Tlvresult`

## 8.531 `pack_uim_SLQSUIMAuthenticate_t` Struct Reference

### Data Fields

- [uim\\_sessionInformation](#) `sessionInfo`
- [uim\\_authenticationData](#) `authData`
- `uint32_t * pIndicationToken`

### 8.531.1 Detailed Description

This structure contains information of the request parameters associated with a `SLQSUIMAuthenticate`.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>authData</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_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.

### 8.531.2 Field Documentation

8.531.2.1 `uim_authenticationData pack_uim_SLQSUIMAuthenticate_t::authData`

8.531.2.2 `uint32_t* pack_uim_SLQSUIMAuthenticate_t::pIndicationToken`

8.531.2.3 `uim_sessionInformation pack_uim_SLQSUIMAuthenticate_t::sessionInfo`

## 8.532 `pack_uim_SLQSUIMDepersonalization_t` Struct Reference

### Data Fields

- [uim\\_depersionalizationInformation](#) `depersonilisationInfo`

### 8.532.1 Detailed Description

This structure contains information of the request parameters associated with a `Depersonalization`.

#### Parameters

<i>depersonilisation-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_depersionalizationInformation</a> for more information.</li> </ul>
-------------------------------	---



### 8.532.2 Field Documentation

8.532.2.1 uim\_depersonalizationInformation pack\_uim\_SLQSUIDepersonalization\_t::depersonalisationInfo

## 8.533 pack\_uim\_SLQSUIMEventRegister\_t Struct Reference

### Data Fields

- uint32\_t [eventMask](#)

### 8.533.1 Detailed Description

This structure contains pack event register parameter.

#### Parameters

<i>eventMask</i>	<ul style="list-style-type: none"> <li>- bit 1 - card status</li> <li>• bit 4 - physical slot status</li> </ul>
------------------	---

### 8.533.2 Field Documentation

8.533.2.1 uint32\_t pack\_uim\_SLQSUIMEventRegister\_t::eventMask

## 8.534 pack\_uim\_SLQSUIMGetConfiguration\_t Struct Reference

### Data Fields

- uint32\_t \* [pConfigurationMask](#)

### 8.534.1 Detailed Description

This structure contains information of the request parameters associated with Get Configuration to get the modem configuration for the UIM module

#### 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.

### 8.534.2 Field Documentation

8.534.2.1 uint32\_t\* pack\_uim\_SLQSUIMGetConfiguration\_t::pConfigurationMask

## 8.535 pack\_uim\_SLQSUIMGetFileAttributes\_t Struct Reference

### Data Fields

- [uim\\_sessionInformation](#) sessionInfo
- [uim\\_fileInfo](#) fileIndex
- [uint32\\_t](#) \* pIndicationToken

### 8.535.1 Detailed Description

This structure contains information of the request parameters associated with a Get File Attributes

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_fileInfo</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.

### 8.535.2 Field Documentation

8.535.2.1 [uim\\_fileInfo](#) pack\_uim\_SLQSUIMGetFileAttributes\_t::fileIndex

8.535.2.2 [uint32\\_t\\*](#) pack\_uim\_SLQSUIMGetFileAttributes\_t::pIndicationToken

8.535.2.3 [uim\\_sessionInformation](#) pack\_uim\_SLQSUIMGetFileAttributes\_t::sessionInfo

## 8.536 pack\_uim\_SLQSUIMGetServiceStatus\_t Struct Reference

### Data Fields

- [uim\\_UIMSessionInformation](#) sessionInfo
- [uint32\\_t](#) capMask

### 8.536.1 Detailed Description

This structure contains information of the request parameters associated with a Get Service Status API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMSessionInformation</a> for more information.</li> </ul>
--------------------	---

<i>capMask</i>	Bitmask of the capabilities the client retrieves from the card <ul style="list-style-type: none"> <li>• Bit 0 - FDN status</li> <li>• Bit 1 - Hidden key PIN status (only for USIM cards)</li> <li>• Bit 2 – Index in the EF-DIR (only for UICC cards)</li> </ul>
----------------	---

**Note**

Using NULL for the pointers would make sure that the parameter is not added to the request.

**8.536.2 Field Documentation**

8.536.2.1 uint32\_t pack\_uim\_SLQSUIMGetServiceStatus\_t::capMask

8.536.2.2 uim\_UIMSessionInformation pack\_uim\_SLQSUIMGetServiceStatus\_t::sessionInfo

**8.537 pack\_uim\_SLQSUIMPowerDown\_t Struct Reference****Data Fields**

- uint8\_t [slot](#)

**8.537.1 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>
-------------	--

**8.537.2 Field Documentation**

8.537.2.1 uint8\_t pack\_uim\_SLQSUIMPowerDown\_t::slot

**8.538 pack\_uim\_SLQSUIMPowerUp\_t Struct Reference****Data Fields**

- uint8\_t [slot](#)
- uint8\_t \* [plgnoreHotSwapSwitch](#)

**8.538.1 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>

## 8.538.2 Field Documentation

8.538.2.1 `uint8_t* pack_uim_SLQSUIMPowerUp_t::plgnoreHotSwapSwitch`8.538.2.2 `uint8_t pack_uim_SLQSUIMPowerUp_t::slot`8.539 `pack_uim_SLQSUIMReadRecord_t` Struct Reference

## Data Fields

- [uim\\_UIMSessionInformation sessionInfo](#)
- [uim\\_fileInfo fileIndex](#)
- [uim\\_readRecordInfo readRecord](#)
- `uint16_t *` [pLastRecord](#)
- `uint32_t *` [pIndicationToken](#)

## 8.539.1 Detailed Description

This structure contains information of the request parameters associated with a Read Record API.

## Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_UIMSessionInformation</a> for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_fileInfo</a> for more information.</li> </ul>
<i>readRecord</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_recordInfo</a> for more information.</li> </ul>
<i>pLastRecord</i>	<ul style="list-style-type: none"> <li>Last record. This value is used to read multiple records at the same time.</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.

## 8.539.2 Field Documentation

8.539.2.1 uim\_fileInfo pack\_uim\_SLQSUIMLReadRecord\_t::fileIndex

8.539.2.2 uint32\_t\* pack\_uim\_SLQSUIMLReadRecord\_t::pIndicationToken

8.539.2.3 uint16\_t\* pack\_uim\_SLQSUIMLReadRecord\_t::pLastRecord

8.539.2.4 uim\_readRecordInfo pack\_uim\_SLQSUIMLReadRecord\_t::readRecord

8.539.2.5 uim\_UIMSessionInformation pack\_uim\_SLQSUIMLReadRecord\_t::sessionInfo

## 8.540 pack\_uim\_SLQSUIMLRefreshComplete\_t Struct Reference

## Data Fields

- [uim\\_sessionInformation sessionInfo](#)
- uint8\_t [refreshComplete](#)

## 8.540.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Complete event.

## Parameters

<i>sessionInfo(-Mandatory)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>refresh-Complete(-Mandatory)</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>

## 8.540.2 Field Documentation

8.540.2.1 uint8\_t pack\_uim\_SLQSUIMLRefreshComplete\_t::refreshComplete

8.540.2.2 uim\_sessionInformation pack\_uim\_SLQSUIMLRefreshComplete\_t::sessionInfo

## 8.541 pack\_uim\_SLQSUIMLRefreshGetLastEvent\_t Struct Reference

## Data Fields

- [uim\\_sessionInformation sessionInfo](#)

## 8.541.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Get Last Event.

## Parameters

<i>sessionInfo</i> (-Mandatory)	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
---------------------------------	--

## 8.541.2 Field Documentation

8.541.2.1 [uim\\_sessionInformation](#) [pack\\_uim\\_SLQSUIRefreshGetLastEvent\\_t::sessionInfo](#)8.542 [pack\\_uim\\_SLQSUIRefreshOK\\_t](#) Struct Reference

## Data Fields

- [uim\\_sessionInformation](#) [sessionInfo](#)
- [uint8\\_t](#) [OKtoRefresh](#)

## 8.542.1 Detailed Description

This structure contains Parameters of the Session Information

## Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• Session Information</li> <li>• See <a href="#">uim_sessionInformation</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>

## 8.542.2 Field Documentation

8.542.2.1 [uint8\\_t](#) [pack\\_uim\\_SLQSUIRefreshOK\\_t::OKtoRefresh](#)8.542.2.2 [uim\\_sessionInformation](#) [pack\\_uim\\_SLQSUIRefreshOK\\_t::sessionInfo](#)8.543 [pack\\_uim\\_SLQSUIRefreshRegister\\_t](#) Struct Reference

## Data Fields

- [uim\\_sessionInformation](#) [sessionInfo](#)
- [uim\\_registerRefresh](#) [regRefresh](#)

## 8.543.1 Detailed Description

This structure contains information of the request parameters associated with a Refresh Register.

## Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• Session Information params</li> <li>• See <a href="#">uim_sessionInformation</a> for more information</li> </ul>
<i>regRefresh</i>	<ul style="list-style-type: none"> <li>• Register Refresh parameters</li> <li>• See <a href="#">uim_registerRefresh</a> for more information</li> </ul>

## 8.543.2 Field Documentation

8.543.2.1 uim\_registerRefresh pack\_uim\_SLQSUIMRefreshRegister\_t::regRefresh

8.543.2.2 uim\_sessionInformation pack\_uim\_SLQSUIMRefreshRegister\_t::sessionInfo

## 8.544 pack\_uim\_SLQSUIMSetServiceStatus\_t Struct Reference

## Data Fields

- [uim\\_UIMSessionInformation](#) sessionInfo
- uint8\_t \* [pFDNStatus](#)

## 8.544.1 Detailed Description

This structure contains information of the request parameters associated with a Set Service Status API.

## Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMSessionInformation</a> for more information.</li> </ul>
<i>pFDNStatus</i>	(Optional) <ul style="list-style-type: none"> <li>• 0: Disables the FDN</li> <li>• 1: Enables the FDN</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

## 8.544.2 Field Documentation

8.544.2.1 uint8\_t\* pack\_uim\_SLQSUIMSetServiceStatus\_t::pFDNStatus

8.544.2.2 uim\_UIMSessionInformation pack\_uim\_SLQSUIMSetServiceStatus\_t::sessionInfo

## 8.545 pack\_uim\_SLQSUIMSwitchSlot\_t Struct Reference

## Data Fields

- uint8\_t [bLogicalSlot](#)
- uint32\_t [ulPhysicalSlot](#)

### 8.545.1 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>

### 8.545.2 Field Documentation

8.545.2.1 `uint8_t pack_uim_SLQSUIMSwitchSlot_t::bLogicalSlot`

8.545.2.2 `uint32_t pack_uim_SLQSUIMSwitchSlot_t::ulPhysicalSlot`

## 8.546 `pack_uim_SLQSUIMWriteRecord_t` Struct Reference

### Data Fields

- [uim\\_UIMSessionInformation sessionInfo](#)
- [uim\\_fileInfo fileIndex](#)
- [uim\\_writeRecordInfo writeRecord](#)
- `uint32_t * pIndicationToken`

### 8.546.1 Detailed Description

This structure contains information of the request parameters associated with a Write Record API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_UIMSessionInformation</a> for more information.</li> </ul>
<i>fileIndex</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_fileInfo</a> for more information.</li> </ul>
<i>writeRecord</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_writeRecordInfo</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.

**8.546.2 Field Documentation**

8.546.2.1 uim\_fileInfo pack\_uim\_SLQSUIWriteRecord\_t::fileIndex

8.546.2.2 uint32\_t\* pack\_uim\_SLQSUIWriteRecord\_t::pIndicationToken

8.546.2.3 uim\_UIMSessionInformation pack\_uim\_SLQSUIWriteRecord\_t::sessionInfo

8.546.2.4 uim\_writeRecordInfo pack\_uim\_SLQSUIWriteRecord\_t::writeRecord

**8.547 pack\_uim\_SLQSUIWriteTransparent\_t Struct Reference****Data Fields**

- [uim\\_UIMSessionInformation sessionInfo](#)
- [uim\\_fileInfo fileIndex](#)
- [uim\\_writeTransparentInfo writeTransparent](#)
- uint32\_t \* [pIndicationToken](#)

**8.547.1 Detailed Description**

This structure contains information of the request parameters associated with a Write Transparent API.

**Parameters**

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMSessionInformation</a> for more information.</li> </ul>
<i>fileId</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_fileInfo</a> for more information.</li> </ul>
<i>writeTransparent</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_writeTransparentInfo</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.

## 8.547.2 Field Documentation

8.547.2.1 `uim_fileInfo` `pack_uim_SLQSUIWriteTransparent_t::fileIndex`

8.547.2.2 `uint32_t*` `pack_uim_SLQSUIWriteTransparent_t::pIndicationToken`

8.547.2.3 `uim_UIMSessionInformation` `pack_uim_SLQSUIWriteTransparent_t::sessionInfo`

8.547.2.4 `uim_writeTransparentInfo` `pack_uim_SLQSUIWriteTransparent_t::writeTransparent`

## 8.548 `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](#)

### 8.548.1 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>sessionInfo</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>

## 8.548.2 Field Documentation

8.548.2.1 uim\_encryptedPIN1 pack\_uim\_UnblockPin\_t::EncryptedPIN1

8.548.2.2 uint32\_t\* pack\_uim\_UnblockPin\_t::pIndicationToken

8.548.2.3 uim\_unblockUIMPIN pack\_uim\_UnblockPin\_t::pinProtection

8.548.2.4 uint8\_t\* pack\_uim\_UnblockPin\_t::pKeyReferenceID

8.548.2.5 uim\_sessionInformation pack\_uim\_UnblockPin\_t::sessionInfo

8.548.2.6 uint16\_t pack\_uim\_UnblockPin\_t::Tlvresult

## 8.549 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](#)

### 8.549.1 Detailed Description

This structure contains information of the request parameters associated with a verify PIN API.

#### Parameters

<i>sessionInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_sessionInformation</a> for more information.</li> </ul>
<i>verifyPIN</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_verifyUIMPIN</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>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.

## 8.549.2 Field Documentation

8.549.2.1 `uim_encryptedPIN1*` `pack_uim_VerifyPin_t::pEncryptedPIN1`

8.549.2.2 `uint32_t*` `pack_uim_VerifyPin_t::pIndicationToken`

8.549.2.3 `uint8_t*` `pack_uim_VerifyPin_t::pKeyReferenceID`

8.549.2.4 `uim_sessionInformation` `pack_uim_VerifyPin_t::sessionInfo`

8.549.2.5 `uint16_t` `pack_uim_VerifyPin_t::Tlvresult`

8.549.2.6 `uim_verifyUIMPIN` `pack_uim_VerifyPin_t::verifyPIN`

## 8.550 `pack_voice_AnswerUSSD_t` Struct Reference

### Data Fields

- `uint8_t *` [pInfo](#)

### 8.550.1 Detailed Description

This structure contains pack anser USSD parameter.

#### Parameters

<i>pInfo</i>	<ul style="list-style-type: none"> <li>• USS information</li> <li>• See <a href="#">voice_USSInfo</a> for more details</li> </ul>
--------------	---

## 8.550.2 Field Documentation

8.550.2.1 `uint8_t*` `pack_voice_AnswerUSSD_t::pInfo`

## 8.551 `pack_voice_OriginateUSSD_t` Struct Reference

### Data Fields

- `uint8_t *` [pInfo](#)

### 8.551.1 Detailed Description

This structure contains pack orginate USSD parameter.

## Parameters

<i>plInfo</i>	<ul style="list-style-type: none"><li>• USS information</li><li>• See <a href="#">voice_USSInfo</a> for more details</li></ul>
---------------	--

## 8.551.2 Field Documentation

8.551.2.1 uint8\_t\* pack\_voice\_OriginateUSSD\_t::plInfo

## 8.552 pack\_voice\_SLQSOriginateUSSD\_t Struct Reference

## Data Fields

- uint8\_t [ussDCS](#)
- uint8\_t [ussLen](#)
- uint8\_t [ussData](#) [182]

## 8.552.1 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>

## 8.552.2 Field Documentation

8.552.2.1 uint8\_t pack\_voice\_SLQSOriginateUSSD\_t::ussData[182]

8.552.2.2 uint8\_t pack\_voice\_SLQSOriginateUSSD\_t::ussDCS

8.552.2.3 uint8\_t pack\_voice\_SLQSOriginateUSSD\_t::ussLen

## 8.553 pack\_voice\_SLQSVoiceALSSelectLine\_t Struct Reference

## Data Fields

- uint8\_t [lineValue](#)

### 8.553.1 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>
------------------	---

### 8.553.2 Field Documentation

8.553.2.1 `uint8_t pack_voice_SLQSVoiceALSSelectLine_t::lineValue`

## 8.554 `pack_voice_SLQSVoiceALSSetLineSwitching_t` Struct Reference

#### Data Fields

- `uint8_t switchOption`

### 8.554.1 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>
---------------------	--

### 8.554.2 Field Documentation

8.554.2.1 `uint8_t pack_voice_SLQSVoiceALSSetLineSwitching_t::switchOption`

## 8.555 `pack_voice_SLQSVoiceAnswerCall_t` Struct Reference

#### Data Fields

- `uint8_t * pCallId`

### 8.555.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

#### Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> <li>Unique call identifier for the call that must be answered.</li> </ul>
----------------	--

## 8.555.2 Field Documentation

8.555.2.1 uint8\_t\* pack\_voice\_SLQSVoiceAnswerCall\_t::pCallId

## 8.556 pack\_voice\_SLQSVoiceBindSubscription\_t Struct Reference

### Data Fields

- uint8\_t [subsType](#)

### 8.556.1 Detailed Description

This structure contains Bind Subscription Information Parameters.

#### Parameters

<i>subsType</i>	<ul style="list-style-type: none"><li>• Subscription Type.<ul style="list-style-type: none"><li>– 0x00 - VOICE_SUBS_TYPE_PRIMARY - Primary</li><li>– 0x01 - VOICE_SUBS_TYPE_SECONDARY - Secondary</li></ul></li></ul>
-----------------	---

## 8.556.2 Field Documentation

8.556.2.1 uint8\_t pack\_voice\_SLQSVoiceBindSubscription\_t::subsType

## 8.557 pack\_voice\_SLQSVoiceBurstDTMF\_t Struct Reference

### Data Fields

- [voice\\_burstDTMFInfo](#) [BurstDTMFInfo](#)
- [voice\\_DTMFLengths](#) \* [pBurstDTMFLengths](#)

### 8.557.1 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="#">voice_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="#">voice_DTMFLengths</a> for more information</li></ul></li><li>• NULL pointer - Invalid data.</li></ul>

## 8.557.2 Field Documentation

8.557.2.1 `voice_burstDTMFInfo` `pack_voice_SLQSVoiceBurstDTMF_t::BurstDTMFInfo`

8.557.2.2 `voice_DTMFLengths*` `pack_voice_SLQSVoiceBurstDTMF_t::pBurstDTMFLengths`

## 8.558 `pack_voice_SLQSVoiceDialCall_t` Struct Reference

### Data Fields

- `uint8_t` `callNumber` [81]
- `uint8_t` \* `pCallType`
- `uint8_t` \* `pCLIRType`
- `voice_UUSInfo` \* `pUUSInfo`
- `voice_CUGInfo` \* `pCUGInfo`
- `uint8_t` \* `pEmergencyCategory`
- `voice_calledPartySubAdd` \* `pCallPartySubAdd`
- `uint8_t` \* `pSvcType`

### 8.558.1 Detailed Description

This structure contains Voice Call Request Parameters

#### Parameters

<i>callNumber</i> [81]	<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</i> (optional)	<ul style="list-style-type: none"> <li>• the type of call to be dialed. <code>CALL_TYPE_VOICE</code> is automatically selected if this parameter is not provided. When <code>CALL_TYPE_NON_STD_OTASP</code> 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 - <code>CALL_TYPE_VOICE</code> - Voice (automatic selection)</li> <li>– 0x01 - <code>CALL_TYPE_VOICE_FORCED</code> - Avoid modem call classification</li> <li>– 0x08 - <code>CALL_TYPE_NON_STD_OTASP</code> - Nonstandard OTASP*</li> <li>– 0x09 - <code>CALL_TYPE_EMERGENCY</code> - Emergency</li> </ul> </li> </ul>
<i>pCLIR-Type</i> (optional)	<ul style="list-style-type: none"> <li>• CLIR type values are: <ul style="list-style-type: none"> <li>– 0x01 - <code>CLIR_SUPPRESSION</code> - Suppression</li> <li>– 0x02 - <code>CLIR_INVOCATION</code> - Invocation</li> </ul> </li> </ul>
<i>pUUSInfo</i> (optional)	<ul style="list-style-type: none"> <li>• Pointer to structure of <code>UUSInfo</code> <ul style="list-style-type: none"> <li>– See <a href="#">voice_UUSInfo</a> for more information</li> </ul> </li> </ul>
<i>pCUG-Info</i> (optional)	<ul style="list-style-type: none"> <li>• Pointer to structure of <code>CUGInfo</code> <ul style="list-style-type: none"> <li>– See <a href="#">voice_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 calledPartySubAdd <ul style="list-style-type: none"> <li>– See <a href="#">voice_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>

## 8.558.2 Field Documentation

8.558.2.1 uint8\_t pack\_voice\_SLQSVoiceDialCall\_t::callNumber[81]

8.558.2.2 voice\_calledPartySubAdd\* pack\_voice\_SLQSVoiceDialCall\_t::pCallPartySubAdd

8.558.2.3 uint8\_t\* pack\_voice\_SLQSVoiceDialCall\_t::pCallType

8.558.2.4 uint8\_t\* pack\_voice\_SLQSVoiceDialCall\_t::pCLIRType

8.558.2.5 voice\_CUGInfo\* pack\_voice\_SLQSVoiceDialCall\_t::pCUGInfo

8.558.2.6 uint8\_t\* pack\_voice\_SLQSVoiceDialCall\_t::pEmergencyCategory

8.558.2.7 uint8\_t\* pack\_voice\_SLQSVoiceDialCall\_t::pSvcType

8.558.2.8 voice\_UUSInfo\* pack\_voice\_SLQSVoiceDialCall\_t::pUUSInfo

## 8.559 pack\_voice\_SLQSVoiceEndCall\_t Struct Reference

### Data Fields

- uint8\_t \* [pCallId](#)

### 8.559.1 Detailed Description

This structure contain pack voice end call parameter.

#### Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> <li>Unique call identifier for the call that must be ended</li> </ul>
----------------	--

### 8.559.2 Field Documentation

8.559.2.1 `uint8_t* pack_voice_SLQSVoiceEndCall_t::pCallId`

## 8.560 `pack_voice_SLQSVoiceGetCallBarring_t` Struct Reference

#### Data Fields

- `uint8_t reason`
- `uint8_t * pSvcClass`

### 8.560.1 Detailed Description

This structure contains Voice Get Call Barring Pack 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>	<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>

### 8.560.2 Field Documentation

8.560.2.1 `uint8_t* pack_voice_SLQSVoiceGetCallBarring_t::pSvcClass`

8.560.2.2 uint8\_t pack\_voice\_SLQSVoiceGetCallBarring\_t::reason

## 8.561 pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t Struct Reference

### Data Fields

- uint8\_t [Reason](#)
- uint8\_t \* [pSvcClass](#)

### 8.561.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Pack 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>

### 8.561.2 Field Documentation

8.561.2.1 uint8\_t\* pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t::pSvcClass

8.561.2.2 uint8\_t pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t::Reason

## 8.562 pack\_voice\_SLQSVoiceGetCallInfo\_t Struct Reference

### Data Fields

- uint8\_t [callID](#)

### 8.562.1 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>
---------------	---

## 8.562.2 Field Documentation

8.562.2.1 uint8\_t pack\_voice\_SLQSVoiceGetCallInfo\_t::callID

## 8.563 pack\_voice\_SLQSVoiceGetCallWaiting\_t Struct Reference

## Data Fields

- uint8\_t \* [pSvcClass](#)

## 8.563.1 Detailed Description

This structure contains Voice Get Call Waiting Response Parameters

## Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> </ul>
------------------	--

## 8.563.2 Field Documentation

8.563.2.1 uint8\_t\* pack\_voice\_SLQSVoiceGetCallWaiting\_t::pSvcClass

## 8.564 pack\_voice\_SLQSVoiceGetConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pAutoAnswer](#)
- uint8\_t \* [pAirTimer](#)
- uint8\_t \* [pRoamTimer](#)
- uint8\_t \* [pTTYMode](#)
- uint8\_t \* [pPrefVoiceSO](#)
- uint8\_t \* [pAMRStatus](#)
- uint8\_t \* [pPrefVoicePrivacy](#)
- uint8\_t \* [pNamID](#)
- uint8\_t \* [pVoiceDomainPref](#)

## 8.564.1 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>pPrefVoiceS-O(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>pNamI-D(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 Domain 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.

## 8.564.2 Field Documentation

## 8.564.2.1 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pAirTimer

- 8.564.2.2 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pAMRStatus
- 8.564.2.3 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pAutoAnswer
- 8.564.2.4 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pNamID
- 8.564.2.5 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pPrefVoicePrivacy
- 8.564.2.6 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pPrefVoiceSO
- 8.564.2.7 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pRoamTimer
- 8.564.2.8 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pTTYMode
- 8.564.2.9 uint8\_t\* pack\_voice\_SLQSVoiceGetConfig\_t::pVoiceDomainPref

## 8.565 pack\_voice\_SLQSVoiceIndicationRegister\_t Struct Reference

### Data Fields

- uint8\_t \* [pRegDTMFEvents](#)
- uint8\_t \* [pRegVoicePrivacyEvents](#)
- uint8\_t \* [pSuppsNotifEvents](#)

### 8.565.1 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.

## 8.565.2 Field Documentation

8.565.2.1 uint8\_t\* pack\_voice\_SLQSVoiceIndicationRegister\_t::pRegDTMFEvents

8.565.2.2 uint8\_t\* pack\_voice\_SLQSVoiceIndicationRegister\_t::pRegVoicePrivacyEvents

8.565.2.3 uint8\_t\* pack\_voice\_SLQSVoiceIndicationRegister\_t::pSuppsNotifEvents

## 8.566 pack\_voice\_SLQSVoiceManageCalls\_t Struct Reference

## Data Fields

- uint8\_t SUPSType
- uint8\_t \* pCallID

## 8.566.1 Detailed Description

This structure contains Manage Calls Information.

## Parameters

<i>SUPSType</i>	<ul style="list-style-type: none"> <li>• Supplementary service type during the call. <ul style="list-style-type: none"> <li>– 0x01 - SUPS_TYPE_RELEASE_HELD_OR_WAITING <ul style="list-style-type: none"> <li>* Release is held or waiting</li> </ul> </li> <li>– 0x02 - SUPS_TYPE_RELEASE_ACTIVE_ACCEPT_HELD_OR_WAITING <ul style="list-style-type: none"> <li>* Release is active and accepting held or waiting</li> </ul> </li> <li>– 0x03 - SUPS_TYPE_HOLD_ACTIVE_ACCEPT_WAITING_OR_HELD <ul style="list-style-type: none"> <li>* Hold is active and accepting waiting or held</li> </ul> </li> <li>– 0x04 - SUPS_TYPE_HOLD_ALL_EXCEPT_SPECIFIED_CALL <ul style="list-style-type: none"> <li>* Hold all calls except a specified one</li> </ul> </li> <li>– 0x05 - SUPS_TYPE_MAKE_CONFERENCE_CALL <ul style="list-style-type: none"> <li>* Make a conference call</li> </ul> </li> <li>– 0x06 - SUPS_TYPE_EXPLICIT_CALL_TRANSFER <ul style="list-style-type: none"> <li>* Explicit call transfer</li> </ul> </li> <li>– 0x07 - SUPS_TYPE_CCBS_ACTIVATION <ul style="list-style-type: none"> <li>* Activate completion of calls to busy subscriber</li> </ul> </li> <li>– 0x08 - SUPS_TYPE_END_ALL_CALLS <ul style="list-style-type: none"> <li>* End all calls</li> </ul> </li> <li>– 0x09 - SUPS_TYPE_RELEASE_SPECIFIED_CALL <ul style="list-style-type: none"> <li>* Release a specified call</li> </ul> </li> </ul> </li> </ul>
<i>pCallID[Optional]</i>	<ul style="list-style-type: none"> <li>• Applicable only for SUPSType 0x04, 0x07, and 0x09</li> <li>• NULL pointer - Invalid data.</li> </ul>

## 8.566.2 Field Documentation

8.566.2.1 `uint8_t*` `pack_voice_SLQSVoiceManageCalls_t::pCallID`

8.566.2.2 `uint8_t` `pack_voice_SLQSVoiceManageCalls_t::SUPSType`

## 8.567 `pack_voice_SLQSVoiceOrigUSSDNoWait_t` Struct Reference

### Data Fields

- struct [voice\\_USSInfo](#) `USSInformation`

### 8.567.1 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="#">voice_USSInfo</a> for more information.</li> </ul>
-----------------------	---

## 8.567.2 Field Documentation

8.567.2.1 `struct voice_USSInfo` `pack_voice_SLQSVoiceOrigUSSDNoWait_t::USSInformation`

## 8.568 `pack_voice_SLQSVoiceSendFlash_t` Struct Reference

### Data Fields

- `uint8_t*` `pCallID`
- `uint8_t*` `pFlashPayLd`
- `uint8_t*` `pFlashType`

### 8.568.1 Detailed Description

This structure contains the flash information associated with a call.

#### Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier associated with the current call.</li> </ul>
<i>pFlashPayLd(optional)</i>	<ul style="list-style-type: none"> <li>• Payload in ASCII to be sent in Flash.</li> <li>• Variable Length, NULL terminated.</li> </ul>
<i>pFlashType(optional)</i>	<ul style="list-style-type: none"> <li>• Flash type. <ul style="list-style-type: none"> <li>– 0 - Simple Flash (default)</li> <li>– 1 - Activate answer hold</li> <li>– 2 - Deactivate answer hold</li> </ul> </li> </ul>



## 8.568.2 Field Documentation

8.568.2.1 uint8\_t\* pack\_voice\_SLQSVoiceSendFlash\_t::pCallID

8.568.2.2 uint8\_t\* pack\_voice\_SLQSVoiceSendFlash\_t::pFlashPayLd

8.568.2.3 uint8\_t\* pack\_voice\_SLQSVoiceSendFlash\_t::pFlashType

## 8.569 pack\_voice\_SLQSVoiceSetCallBarringPassword\_t Struct Reference

### Data Fields

- uint8\_t [Reason](#)
- uint8\_t [oldPasswd](#) [4]
- uint8\_t [newPasswd](#) [4]
- uint8\_t [newPasswdAgain](#) [4]

### 8.569.1 Detailed Description

This structure contains Voice Set Call Barring Password Pack 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_OUTGOINGINTEXTOTHOME - 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[BARRING_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[BARRING_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[BARRING_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>

## 8.569.2 Field Documentation

8.569.2.1 `uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswd[4]`

8.569.2.2 `uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::newPasswdAgain[4]`

8.569.2.3 `uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::oldPasswd[4]`

8.569.2.4 `uint8_t pack_voice_SLQSVoiceSetCallBarringPassword_t::Reason`

## 8.570 `pack_voice_SLQSVoiceSetConfig_t` Struct Reference

### Data Fields

- `uint8_t * pAutoAnswer`
- `voice_airTimer * pAirTimerConfig`
- `voice_roamTimer * pRoamTimerConfig`
- `uint8_t * pTTYMode`
- `voice_prefVoiceSO * pPrefVoiceSO`
- `uint8_t * pPrefVoiceDomain`

### 8.570.1 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="#">voice_airTimer</a> for more information</li> </ul>
<i>pRoamTimerConfig</i>	<ul style="list-style-type: none"> <li>• Value specified is written to NV_ROAM_CNT_I. (optional)</li> <li>• See <a href="#">voice_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="#">voice_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.

**8.570.2 Field Documentation**

8.570.2.1 `voice_airTimer*` `pack_voice_SLQSVoiceSetConfig_t::pAirTimerConfig`

8.570.2.2 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pAutoAnswer`

8.570.2.3 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceDomain`

8.570.2.4 `voice_prefVoiceSO*` `pack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSO`

8.570.2.5 `voice_roamTimer*` `pack_voice_SLQSVoiceSetConfig_t::pRoamTimerConfig`

8.570.2.6 `uint8_t*` `pack_voice_SLQSVoiceSetConfig_t::pTTYMode`

**8.571 pack\_voice\_SLQSVoiceSetPreferredPrivacy\_t Struct Reference****Data Fields**

- `uint8_t` [privacyPref](#)

**8.571.1 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>
--------------------	--

**8.571.2 Field Documentation**

8.571.2.1 `uint8_t` `pack_voice_SLQSVoiceSetPreferredPrivacy_t::privacyPref`

**8.572 pack\_voice\_SLQSVoiceSetSUPSService\_t Struct Reference**

## Data Fields

- uint8\_t [voiceSvc](#)
- uint8\_t [reason](#)
- uint8\_t \* [pServiceClass](#)
- uint8\_t \* [pCallBarringPasswd](#)
- uint8\_t \* [pCallForwardingNumber](#)
- uint8\_t \* [pTimerVal](#)
- [voice\\_callFwdTypeAndPlan](#) \* [pCallFwdTypeAndPlan](#)

### 8.572.1 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="#">liteServiceClassInformation</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>pCallFwdTypeAndPlan</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="#">voice_callFwdTypeAndPlan</a> for more information</li> </ul> </li> </ul>
----------------------------	--

## 8.572.2 Field Documentation

8.572.2.1 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallBarringPasswd`

8.572.2.2 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pCallForwardingNumber`

8.572.2.3 `voice_callFwdTypeAndPlan* pack_voice_SLQSVoiceSetSUPSService_t::pCallFwdTypeAndPlan`

8.572.2.4 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pServiceClass`

8.572.2.5 `uint8_t* pack_voice_SLQSVoiceSetSUPSService_t::pTimerVal`

8.572.2.6 `uint8_t pack_voice_SLQSVoiceSetSUPSService_t::reason`

8.572.2.7 `uint8_t pack_voice_SLQSVoiceSetSUPSService_t::voiceSvc`

## 8.573 `pack_voice_SLQSVoiceStartContDTMF_t` Struct Reference

### Data Fields

- `uint8_t *` [pCallID](#)
- `uint8_t` [DTMFdigit](#)

### 8.573.1 Detailed Description

This structure contains parameters of continuous DTMF

#### Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> <li>Call ID associated with call on which the DTMF information has to be sent. Start continuous DTMF request is sent to the current active/alerting call when pCallId is set to 0xFF.</li> <li>This is IN/OUT parameter, value passed by user will be packed in request and value received from the device would be returned to the user.</li> <li>If the call ID value received is 0, no value has been returned by the device</li> </ul>
<i>DTMFdigit</i>	<ul style="list-style-type: none"> <li>DTMF digit in ASCII.</li> </ul>

## 8.573.2 Field Documentation

8.573.2.1 `uint8_t pack_voice_SLQSVoiceStartContDTMF_t::DTMFdigit`

8.573.2.2 uint8\_t\* pack\_voice\_SLQSVoiceStartContDTMF\_t::pCallID

## 8.574 pack\_voice\_SLQSVoiceStopContDTMF\_t Struct Reference

### Data Fields

- uint8\_t [callID](#)

### 8.574.1 Detailed Description

This structure contains parameters of stop continuous DTMF

#### Parameters

<i>pCallID</i>	<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>
----------------	--

### 8.574.2 Field Documentation

8.574.2.1 uint8\_t pack\_voice\_SLQSVoiceStopContDTMF\_t::callID

## 8.575 pack\_wds\_DHCPv4ClientLeaseChange\_t Struct Reference

### Data Fields

- uint8\_t \* [pEnableNotification](#)

### 8.575.1 Detailed Description

WDS SWI DHCPv4 Client Lease Change Structure

#### Parameters

<i>pEnable-Notification</i>	<ul style="list-style-type: none"> <li>• Enable Notification or not</li> </ul>
-----------------------------	--

### 8.575.2 Field Documentation

8.575.2.1 uint8\_t\* pack\_wds\_DHCPv4ClientLeaseChange\_t::pEnableNotification

## 8.576 pack\_wds\_GetDefaultProfile\_t Struct Reference

### Data Fields

- uint32\_t [profiletype](#)

### 8.576.1 Detailed Description

This structure contains pack get default profile information.

#### Parameters

<i>profiletype</i>	profile type <ul style="list-style-type: none"> <li>• 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP</li> <li>• 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2</li> <li>• 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC</li> </ul>
--------------------	---

### 8.576.2 Field Documentation

8.576.2.1 `uint32_t pack_wds_GetDefaultProfile_t::profiletype`

## 8.577 `pack_wds_GetDefaultProfileNum_t` Struct Reference

#### Data Fields

- `uint8_t type`
- `uint8_t family`

### 8.577.1 Detailed Description

This structure contains pack Get Default Profile Number information.

#### Parameters

<i>type</i>	profile type <ul style="list-style-type: none"> <li>• 0 - 3GPP</li> <li>• 1 - 3GPP2</li> </ul>
<i>family</i>	profile family <ul style="list-style-type: none"> <li>• 0 - Embedded</li> <li>• 1 - Tethered</li> </ul>

### 8.577.2 Field Documentation

8.577.2.1 `uint8_t pack_wds_GetDefaultProfileNum_t::family`

8.577.2.2 `uint8_t pack_wds_GetDefaultProfileNum_t::type`

## 8.578 `pack_wds_GetDefaultProfileV2_t` Struct Reference

#### Data Fields

- `uint32_t profiletype`



### 8.578.1 Detailed Description

This structure contains pack get default profile information.

#### Parameters

<i>profiletype</i>	profile type <ul style="list-style-type: none"> <li>• 0 - WDS_PROFILE_TYPE_3GPP (0x00) - 3GPP</li> <li>• 1 - WDS_PROFILE_TYPE_3GPP (0x01) - 3GPP2</li> <li>• 2 - WDS_PROFILE_TYPE_EPC (0x02) - EPC</li> </ul>
--------------------	---

### 8.578.2 Field Documentation

8.578.2.1 uint32\_t pack\_wds\_GetDefaultProfileV2\_t::profiletype

## 8.579 pack\_wds\_GetDormancyState\_t Struct Reference

### 8.579.1 Detailed Description

This structure contains pack get dormancy state information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>• this is a dummy structure</li> </ul>
-------------	---

## 8.580 pack\_wds\_GetLastMobileIPError\_t Struct Reference

### 8.580.1 Detailed Description

This structure contains pack get last mobile IP error information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>• this is a dummy structure</li> </ul>
-------------	---

## 8.581 pack\_wds\_GetMobileIP\_t Struct Reference

### 8.581.1 Detailed Description

This structure contains pack get mobile IP information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>• this is a dummy structure</li> </ul>
-------------	---

## 8.582 pack\_wds\_GetMobileIPProfile\_t Struct Reference

### Data Fields

- uint8\_t [index](#)

### 8.582.1 Detailed Description

This structure contains pack get mobile IP profile information.

#### Parameters

<i>index</i>	<ul style="list-style-type: none"> <li>• Mobile IP profile ID</li> </ul>
--------------	--

### 8.582.2 Field Documentation

8.582.2.1 uint8\_t pack\_wds\_GetMobileIPProfile\_t::index

## 8.583 pack\_wds\_GetPacketStatistics\_t Struct Reference

### Data Fields

- uint32\_t \* [pStatMask](#)

### 8.583.1 Detailed Description

This structure contains pack get packet statistics information.

#### Parameters

<i>pStatMask</i>	<ul style="list-style-type: none"> <li>• Packet Statistics 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> </ul>
------------------	--

### 8.583.2 Field Documentation

8.583.2.1 uint32\_t\* pack\_wds\_GetPacketStatistics\_t::pStatMask

## 8.584 pack\_wds\_GetPacketStatus\_t Struct Reference

## Data Fields

- uint32\_t [statmask](#)

### 8.584.1 Detailed Description

This structure contains pack get packet status information.

#### Parameters

<i>statmask</i>	packet statistics 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> <li>• 0x00000100 - Tx packets dropped</li> <li>• 0x00000200 - Rx packets dropped</li> </ul>
-----------------	---

### 8.584.2 Field Documentation

8.584.2.1 uint32\_t pack\_wds\_GetPacketStatus\_t::statmask

## 8.585 pack\_wds\_GetSessionDuration\_t Struct Reference

### 8.585.1 Detailed Description

This structure contains pack get session duration information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>• this is a dummy structure</li> </ul>
-------------	---

## 8.586 pack\_wds\_RMSetTransferStatistics\_t Struct Reference

## Data Fields

- [rmTrasnferStaticsReq](#) RmTrasnferStaticsReq

### 8.586.1 Detailed Description

This structure contains pack fetch current data system transfer statistics information.

## Parameters

<i>RmTrasfer-StaticsReq[IN]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">rmTransferStaticsReq</a> for more information</li> </ul>
---------------------------------	---

## 8.586.2 Field Documentation

8.586.2.1 `rmTransferStaticsReq pack_wds_RMSetTransferStatistics_t::RmTrasferStaticsReq`8.587 `pack_wds_SetAutoconnect_t` Struct Reference

## Data Fields

- `uint8_t` [acsetting](#)
- `uint8_t` [acroamsetting](#)

## 8.587.1 Detailed Description

auto connect data session parameters.

## Parameters

<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>

## 8.587.2 Field Documentation

8.587.2.1 `uint8_t pack_wds_SetAutoconnect_t::acroamsetting`8.587.2.2 `uint8_t pack_wds_SetAutoconnect_t::acsetting`8.588 `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

### 8.588.1 Detailed Description

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 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>
<i>pdpType</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> <li>– 1 - PDP-PPP</li> <li>– 2 - PDP-IPv6</li> <li>– 3 - PDP-IPv4v6</li> </ul> </li> </ul>
<i>ipAddress</i>	<ul style="list-style-type: none"> <li>• Preferred IPv4 addr to be assigned to device (optional)</li> </ul>
<i>primaryDNS</i>	<ul style="list-style-type: none"> <li>• Primary DNS ipv4 address preference (optional)</li> </ul>
<i>secondaryDNS</i>	<ul style="list-style-type: none"> <li>• Secondary DNS ipv4 address preference (optional)</li> </ul>
<i>authentication</i>	<ul style="list-style-type: none"> <li>• Bitmap that indicates authentication algorithm preference (optional) <ul style="list-style-type: none"> <li>– 0x00000001 - PAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– 0x00000002 - CHAP preference <ul style="list-style-type: none"> <li>* 0 - Never performed</li> <li>* 1 - May be performed</li> </ul> </li> <li>– All other bits are reserved and must be set to 0</li> <li>– If more than 1 bit is set, then device decides which authentication procedure is performed while setting up data session e.g. the device may have a policy to select the most secure authentication mechanism.</li> </ul> </li> </ul>
<i>pName</i>	<ul style="list-style-type: none"> <li>• profile Name (optional)</li> </ul>
<i>pUsername</i>	<ul style="list-style-type: none"> <li>• Username used during network authentication (optional)</li> </ul>

<i>pAPNName</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>pPassword</i>	<ul style="list-style-type: none"> <li>Password used during network authentication (optional)</li> </ul>

## 8.588.2 Field Documentation

8.588.2.1 uint32\_t pack\_wds\_SetDefaultProfile\_t::authentication

8.588.2.2 uint32\_t pack\_wds\_SetDefaultProfile\_t::ipAddress

8.588.2.3 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pApnname

8.588.2.4 uint32\_t pack\_wds\_SetDefaultProfile\_t::pdpType

8.588.2.5 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pName

8.588.2.6 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pPassword

8.588.2.7 uint32\_t pack\_wds\_SetDefaultProfile\_t::primaryDNS

8.588.2.8 uint32\_t pack\_wds\_SetDefaultProfile\_t::profileType

8.588.2.9 uint8\_t\* pack\_wds\_SetDefaultProfile\_t::pUsername

8.588.2.10 uint32\_t pack\_wds\_SetDefaultProfile\_t::secondaryDNS

## 8.589 pack\_wds\_SetDefaultProfileNum\_t Struct Reference

### Data Fields

- uint8\_t [type](#)
- uint8\_t [family](#)
- uint8\_t [index](#)

### 8.589.1 Detailed Description

This structure to hold Set default profile number

#### Parameters

<i>type</i>	Identifies the technology type of the profile <ul style="list-style-type: none"> <li>0 - 3GPP</li> <li>1 - 3GPP2</li> </ul>
<i>family</i>	Identifies the family of profile <ul style="list-style-type: none"> <li>0 - Embedded</li> <li>1 - Tethered</li> </ul>
<i>index</i>	Profile number to be set as default profile.

## 8.589.2 Field Documentation

8.589.2.1 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::family

8.589.2.2 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::index

8.589.2.3 uint8\_t pack\_wds\_SetDefaultProfileNum\_t::type

## 8.590 pack\_wds\_SetMobileIP\_t Struct Reference

### Data Fields

- uint32\_t [mode](#)

### 8.590.1 Detailed Description

This structure contains set mobile IP pack information.

#### Parameters

<i>mode</i>	<ul style="list-style-type: none"><li>Mobile IP setting<ul style="list-style-type: none"><li>0 - Mobile IP off (simple IP only)</li><li>1 - Mobile IP preferred</li><li>2 - Mobile IP only</li></ul></li></ul>
-------------	--

## 8.590.2 Field Documentation

8.590.2.1 uint32\_t pack\_wds\_SetMobileIP\_t::mode

## 8.591 pack\_wds\_SetMobileIPParameters\_t Struct Reference

### Data Fields

- char \* [pSPC](#)
- uint32\_t \* [pMode](#)
- uint8\_t \* [pRetryLimit](#)
- uint8\_t \* [pRetryInterval](#)
- uint8\_t \* [pReRegPeriod](#)
- uint8\_t \* [pReRegTraffic](#)
- uint8\_t \* [pHAAAuthenticator](#)
- uint8\_t \* [pHA2002bis](#)

### 8.591.1 Detailed Description

Mobile IP parameters information.

#### Parameters

<i>pSPC</i>	<ul style="list-style-type: none"><li>NULL-terminated string representing six digit service programming code.</li></ul>
-------------	---

<i>pMode</i>	<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>	<ul style="list-style-type: none"> <li>• Registration retry attempt limit (optional)</li> </ul>
<i>pRetryInterval</i>	<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>	<ul style="list-style-type: none"> <li>• Period (in minutes) to attempt re-registration before current registration expires (optional)</li> </ul>
<i>pReRegTraffic</i>	<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>	<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>	<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>

## 8.591.2 Field Documentation

8.591.2.1 `uint8_t* pack_wds_SetMobileIPParameters_t::pHA2002bis`

8.591.2.2 `uint8_t* pack_wds_SetMobileIPParameters_t::pHAAuthenticator`

8.591.2.3 `uint32_t* pack_wds_SetMobileIPParameters_t::pMode`

8.591.2.4 `uint8_t* pack_wds_SetMobileIPParameters_t::pReRegPeriod`

8.591.2.5 `uint8_t* pack_wds_SetMobileIPParameters_t::pReRegTraffic`

8.591.2.6 `uint8_t* pack_wds_SetMobileIPParameters_t::pRetryInterval`

8.591.2.7 `uint8_t* pack_wds_SetMobileIPParameters_t::pRetryLimit`

8.591.2.8 `char* pack_wds_SetMobileIPParameters_t::pSPC`

## 8.592 `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](#)

## 8.592.1 Detailed Description

Sets the mobile IP parameters.

## Parameters

<i>spc</i>	<ul style="list-style-type: none"> <li>• Six digit service programming code string</li> </ul>
<i>index</i>	<ul style="list-style-type: none"> <li>• Index of the profile to modify</li> </ul>
<i>pEnabled</i>	<ul style="list-style-type: none"> <li>• (Optional) Enable profile? 0 - Disabled Nonzero - Enabled</li> </ul>
<i>pAddress</i>	<ul style="list-style-type: none"> <li>• (Optional) Home IPv4 address</li> </ul>
<i>pPrimaryHA</i>	<ul style="list-style-type: none"> <li>• (Optional) Primary home agent IPv4 address</li> </ul>
<i>pSecondaryHA</i>	<ul style="list-style-type: none"> <li>• (Optional) Secondary home agent IPv4 address</li> </ul>
<i>pRevTunneling</i>	<ul style="list-style-type: none"> <li>• (Optional) Enable reverse tunneling? 0 - Disabled Nonzero - Enabled</li> </ul>
<i>pNAI</i>	<ul style="list-style-type: none"> <li>• (Optional) Network access identifier string</li> </ul>
<i>pHASPI</i>	<ul style="list-style-type: none"> <li>• (Optional) Home agent security parameter index</li> </ul>
<i>pAAASPI</i>	<ul style="list-style-type: none"> <li>• (Optional) AAA server security parameter index</li> </ul>
<i>pMNHA</i>	<ul style="list-style-type: none"> <li>• (Optional) MN-HA key string</li> </ul>

<i>pMNAAA</i>	<ul style="list-style-type: none"> <li>• (Optional) MN-AAA key string</li> </ul>
---------------	--

## 8.592.2 Field Documentation

8.592.2.1 `uint8_t pack_wds_SetMobileIPProfile_t::index`

8.592.2.2 `uint32_t* pack_wds_SetMobileIPProfile_t::pAAASPI`

8.592.2.3 `uint32_t* pack_wds_SetMobileIPProfile_t::pAddress`

8.592.2.4 `uint8_t* pack_wds_SetMobileIPProfile_t::pEnabled`

8.592.2.5 `uint32_t* pack_wds_SetMobileIPProfile_t::pHASPI`

8.592.2.6 `int8_t* pack_wds_SetMobileIPProfile_t::pMNAAA`

8.592.2.7 `int8_t* pack_wds_SetMobileIPProfile_t::pMNHA`

8.592.2.8 `int8_t* pack_wds_SetMobileIPProfile_t::pNAI`

8.592.2.9 `uint32_t* pack_wds_SetMobileIPProfile_t::pPrimaryHA`

8.592.2.10 `uint8_t* pack_wds_SetMobileIPProfile_t::pRevTunneling`

8.592.2.11 `uint32_t* pack_wds_SetMobileIPProfile_t::pSecondaryHA`

8.592.2.12 `int8_t pack_wds_SetMobileIPProfile_t::spc[10]`

## 8.593 `pack_wds_SLQSCreateProfile_t` Struct Reference

### Data Fields

- `uint8_t * pProfileId`
- `uint8_t * pProfileType`
- `wds_profileInfo * pCurProfile`

### 8.593.1 Detailed Description

This structure contains pack create profile.

#### Parameters

<i>pProfileId</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>pProfileType</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>pCurProfile</i>	<ul style="list-style-type: none"> <li>union of 3GPP and 3GPP2 profile</li> <li>See <a href="#">wds_profileInfo</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

**8.593.2 Field Documentation**8.593.2.1 `wds_profileInfo*` `pack_wds_SLQSCreateProfile_t::pCurProfile`8.593.2.2 `uint8_t*` `pack_wds_SLQSCreateProfile_t::pProfileId`8.593.2.3 `uint8_t*` `pack_wds_SLQSCreateProfile_t::pProfileType`**8.594 pack\_wds\_SLQSDestroyProfile\_t Struct Reference****Data Fields**

- `uint8_t` [profileType](#)
- `uint8_t` [profileIndex](#)

**8.594.1 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>

**8.594.2 Field Documentation**

8.594.2.1 `uint8_t pack_wds_SLQSDDeleteProfile_t::profileIndex`

8.594.2.2 `uint8_t pack_wds_SLQSDDeleteProfile_t::profileType`

## 8.595 `pack_wds_SLQSGetCurrDataSystemStat_t` Struct Reference

### 8.595.1 Detailed Description

This structure contains pack get current data system state information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>this is a dummy structure</li> </ul>
-------------	---

## 8.596 `pack_wds_SLQSGetDataBearerTechnology_t` Struct Reference

### 8.596.1 Detailed Description

This structure contains pack get data bearer technology information.

#### Parameters

<i>NULL</i>	<ul style="list-style-type: none"> <li>this is a dummy structure</li> </ul>
-------------	---

## 8.597 `pack_wds_SLQSGetDUNCallInfo_t` Struct Reference

### Data Fields

- `uint32_t` [Mask](#)
- `uint8_t *` [pReportConnStatus](#)
- `wds_transferStatInd *` [pTransferStatInd](#)
- `uint8_t *` [pReportDormStatus](#)
- `uint8_t *` [pReportDataBearerTech](#)
- `uint8_t *` [pReportChannelRate](#)

### 8.597.1 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="#">wds_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>

## 8.597.2 Field Documentation

8.597.2.1 uint32\_t pack\_wds\_SLQSGetDUNCallInfo\_t::Mask

8.597.2.2 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportChannelRate

8.597.2.3 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportConnStatus

8.597.2.4 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDataBearerTech

8.597.2.5 uint8\_t\* pack\_wds\_SLQSGetDUNCallInfo\_t::pReportDormStatus

8.597.2.6 wds\_transferStatInd\* pack\_wds\_SLQSGetDUNCallInfo\_t::pTransferStatInd

## 8.598 pack\_wds\_SLQSGetProfileSettings\_t Struct Reference

### Data Fields

- uint8\_t [ProfileId](#)
- uint8\_t [ProfileType](#)

### 8.598.1 Detailed Description

This structure contains pack get profile settings information.

#### 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

### 8.598.2 Field Documentation

8.598.2.1 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileId

8.598.2.2 uint8\_t pack\_wds\_SLQSGetProfileSettings\_t::ProfileType

## 8.599 pack\_wds\_SLQSGetRuntimeSettings\_t Struct Reference

### Data Fields

- uint32\_t \* [pReqSettings](#)

### 8.599.1 Detailed Description

This structure contains pack get runtime settings information

## 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 (Not Supported on MC/EM73xx)</li> </ul> </li> </ul>
---------------------	---

## 8.599.2 Field Documentation

8.599.2.1 uint32\_t\* pack\_wds\_SLQSGetRuntimeSettings\_t::pReqSettings

## 8.600 pack\_wds\_SLQSModifyProfile\_t Struct Reference

## Data Fields

- uint8\_t \* [pProfileId](#)
- uint8\_t \* [pProfileType](#)
- [wds\\_profileInfo](#) curProfile

## 8.600.1 Detailed Description

This structure contains pack modify profile information.

## Parameters

<i>pProfileId</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>pProfileType</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> <li>• See <a href="#">wds_profileInfo</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

## 8.600.2 Field Documentation

8.600.2.1 **wds\_profileInfo** pack\_wds\_SLQSMModifyProfile\_t::curProfile

8.600.2.2 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileId

8.600.2.3 uint8\_t\* pack\_wds\_SLQSMModifyProfile\_t::pProfileType

## 8.601 pack\_wds\_SLQSSet3GPPConfigItem\_t Struct Reference

## Data Fields

- uint16\_t \* [pLTEAttachProfile](#)
- uint16\_t \* [pProfileList](#)
- uint8\_t \* [pDefaultPDNEnabled](#)
- uint8\_t \* [p3gppRelease](#)
- uint16\_t \* [pLTEAttachProfileList](#)
- uint16\_t [LTEAttachProfileListLen](#)

## 8.601.1 Detailed Description

This structure contains pack set 3GPP configuration item information.

## Parameters

<i>pLTEAttach-Profile</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <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> </ul> </li> <li>• This setting is deprecated on MC/EM74xx</li> </ul>
---------------------------	---



<i>ProfileList</i>	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 SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present</li> </ul>
<i>pDefaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>• Optional parameter             <ul style="list-style-type: none"> <li>– 0 - disabled</li> <li>– 1 - enabled</li> </ul> </li> </ul>
<i>p3gppRelease</i>	3GPP release <ul style="list-style-type: none"> <li>• Optional parameter             <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>• In 9x30 and onwards             <ul style="list-style-type: none"> <li>– 5 - Release 9</li> <li>– 6 - Release 10</li> <li>– 7 - Release 11</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>– 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>

## 8.601.2 Field Documentation

8.601.2.1 uint16\_t pack\_wds\_SLQSSet3GPPConfigItem\_t::LTEAttachProfileListLen

8.601.2.2 uint8\_t\* pack\_wds\_SLQSSet3GPPConfigItem\_t::p3gppRelease

8.601.2.3 uint8\_t\* pack\_wds\_SLQSSet3GPPConfigItem\_t::pDefaultPDNEnabled

8.601.2.4 uint16\_t\* pack\_wds\_SLQSSet3GPPConfigItem\_t::pLTEAttachProfile

8.601.2.5 uint16\_t\* pack\_wds\_SLQSSet3GPPConfigItem\_t::pLTEAttachProfileList

8.601.2.6 `uint16_t* pack_wds_SLQSSet3GPPConfigItem_t::pProfileList`

## 8.602 `pack_wds_SLQSSetIPFamilyPreference_t` Struct Reference

### Data Fields

- `uint8_t` [IPFamilyPreference](#)

### 8.602.1 Detailed Description

This structure contains pack set IP family preference information

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

### 8.602.2 Field Documentation

8.602.2.1 `uint8_t pack_wds_SLQSSetIPFamilyPreference_t::IPFamilyPreference`

## 8.603 `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` [dataBearerTechExt](#)
- `uint8_t` [interval](#)

### 8.603.1 Detailed Description

This structure contains set WDS event callback information.

#### 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>dataBearerTech-Ext</i>	data Bearer Technology Extended
<i>interval</i>	interval

### 8.603.2 Field Documentation

8.603.2.1 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::currentDataBearer

8.603.2.2 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataBearer

8.603.2.3 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataBearerTechExt

8.603.2.4 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dataSystemStatus

8.603.2.5 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::dormancyStatus

8.603.2.6 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::interval

8.603.2.7 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::mobileIP

8.603.2.8 uint8\_t pack\_wds\_SLQSSetWdsEventCallback\_t::transferStats

## 8.604 pack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

### Data Fields

- [wdsDhcpv4ProfileId](#) \* [pProfileId](#)

### 8.604.1 Detailed Description

This structure contain get DHCPv4 client configure.

#### Parameters

<i>pProfileId</i>	pointer to Profile Id structure <ul style="list-style-type: none"> <li>• See <a href="#">wdsDhcpv4ProfileId</a></li> </ul>
-------------------	--

### 8.604.2 Field Documentation

8.604.2.1 wdsDhcpv4ProfileId\* pack\_wds\_SLQSSetDHCPv4ClientConfig\_t::pProfileId

## 8.605 pack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

### Data Fields

- [wds\\_DHCPv4ProfileId](#) \* [pProfileId](#)
- [wds\\_DHCPv4HWConfig](#) \* [pHwConfig](#)
- [wds\\_DHCPv4OptionList](#) \* [pRequestOptionList](#)

### 8.605.1 Detailed Description

WDS SWI DHCPv4 Config Structure

## Parameters

<i>pProfileId</i>	<ul style="list-style-type: none"> <li>• pointer to Profile Id structure</li> </ul>
<i>pHWConfig</i>	<ul style="list-style-type: none"> <li>• pointer to HW Config structure</li> </ul>
<i>pRequestOption-List</i>	<ul style="list-style-type: none"> <li>• pointer to Option List structure to be sent in DHCP request</li> </ul>

## 8.605.2 Field Documentation

8.605.2.1 wds\_DHCPv4HWConfig\* pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t::pHwConfig

8.605.2.2 wds\_DHCPv4ProfileId\* pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t::pProfileId

8.605.2.3 wds\_DHCPv4OptionList\* pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t::pRequestOptionList

## 8.606 pack\_wds\_SLQSSSetLoopback\_t Struct Reference

## Data Fields

- uint8\_t [loopbackMode](#)
- uint8\_t [loopbackMultiplier](#)

## 8.606.1 Detailed Description

This structure contains pack set loopback information.

## Parameters

<i>loopbackMode</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>loopback-Multiplier</i>	<ul style="list-style-type: none"> <li>• Loopback multiplier. Number of downlink bytes to send for each uplink byte.</li> </ul>

## 8.606.2 Field Documentation

8.606.2.1 uint8\_t pack\_wds\_SLQSSSetLoopback\_t::loopbackMode

8.606.2.2 uint8\_t pack\_wds\_SLQSSSetLoopback\_t::loopbackMultiplier

## 8.607 pack\_wds\_SLQSSStartDataSession\_t Struct Reference

## Data Fields

- uint8\_t \* [pTech](#)

- uint32\_t \* pprofileid3gpp
- uint32\_t \* pprofileid3gpp2
- uint32\_t \* pAuth
- char \* pUser
- char \* pPass

### 8.607.1 Detailed Description

This structure contains pack Start Data Session Information.

#### 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 3GPP2 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>

### 8.607.2 Field Documentation

8.607.2.1 uint32\_t\* pack\_wds\_SLQSStartDataSession\_t::pAuth

8.607.2.2 char\* pack\_wds\_SLQSStartDataSession\_t::pPass

8.607.2.3 uint32\_t\* pack\_wds\_SLQSStartDataSession\_t::pprofileid3gpp

8.607.2.4 uint32\_t\* pack\_wds\_SLQSStartDataSession\_t::pprofileid3gpp2

8.607.2.5 uint8\_t\* pack\_wds\_SLQSStartDataSession\_t::pTech

8.607.2.6 char\* pack\_wds\_SLQSSstartDataSession\_t::pUser

## 8.608 pack\_wds\_SLQSSstopDataSession\_t Struct Reference

### Data Fields

- uint32\_t \* [psid](#)

### 8.608.1 Detailed Description

This structure contains pack stop data session information.

#### Parameters

<i>psid</i>	session id
-------------	------------

### 8.608.2 Field Documentation

8.608.2.1 uint32\_t\* pack\_wds\_SLQSSstopDataSession\_t::psid

## 8.609 pack\_wds\_SLQSSwiProfileChangeCallback\_t Struct Reference

### Data Fields

- uint8\_t \* [pProfileChangeInd](#)

### 8.609.1 Detailed Description

This structure contains Profile Change Enable/Disable Notification Parameter

#### Parameters

<i>pProfileChangeInd</i>	<ul style="list-style-type: none"> <li>• Enable/Disable Notification</li> </ul>
--------------------------	---

### 8.609.2 Field Documentation

8.609.2.1 uint8\_t\* pack\_wds\_SLQSSwiProfileChangeCallback\_t::pProfileChangeInd

## 8.610 pack\_wds\_SLQSWdsSetEventReport\_t Struct Reference

### Data Fields

- uint8\_t \* [pCurrChannelRateInd](#)
- [wds\\_TrStatInd](#) \* [pTransferStatInd](#)
- uint8\_t \* [pDataBearerTechInd](#)
- uint8\_t \* [pDormancyStatusInd](#)
- uint8\_t \* [pMIPStatusInd](#)
- uint8\_t \* [pCurrDataBearerTechInd](#)
- uint8\_t \* [pDataCallStatusChangeInd](#)
- uint8\_t \* [pCurrPrefDataSysInd](#)

- uint8\_t \* [pEVDOPageMonPerChangeInd](#)
- uint8\_t \* [pDataSystemStatusChangeInd](#)

### 8.610.1 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="#">wds_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.

**8.610.2 Field Documentation**

8.610.2.1 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrChannelRateInd`

8.610.2.2 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrDataBearerTechInd`

8.610.2.3 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pCurrPrefDataSysInd`

8.610.2.4 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataBearerTechInd`

8.610.2.5 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataCallStatusChangeInd`

8.610.2.6 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDataSystemStatusChangeInd`

8.610.2.7 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pDormancyStatusInd`

8.610.2.8 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pEVDOPageMonPerChangeInd`

8.610.2.9 `uint8_t* pack_wds_SLQSWdsSetEventReport_t::pMIPStatusInd`

8.610.2.10 `wds_TrStatInd* pack_wds_SLQSWdsSetEventReport_t::pTransferStatInd`

**8.611 pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t Struct Reference****Data Fields**

- `uint8_t contextId`
- `uint8_t contextType`

**8.611.1 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>contextType</i>	<ul style="list-style-type: none"> <li>• Identifies technology type             <ul style="list-style-type: none"> <li>– 0 - 3GPP</li> <li>– 1 - 3GPP2</li> </ul> </li> </ul>



### 8.611.2 Field Documentation

8.611.2.1 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.611.2.2 `uint8_t` `pack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextType`

## 8.612 PackCreateProfileOut Struct Reference

### Data Fields

- `uint8_t` [ProfileType](#)
- `uint8_t` [ProfileIndex](#)
- `uint16_t` [ExtErrorCode](#)

### 8.612.1 Detailed Description

This 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ProfileIndex</i>	<ul style="list-style-type: none"> <li>• Index identifying the profile that was created</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ExtErrorCode</i>	<ul style="list-style-type: none"> <li>• The extended error code received from DS Profile subsystem</li> <li>• Bit to check in ParamPresenceMask - <b>224</b></li> </ul>

### 8.612.2 Field Documentation

8.612.2.1 `uint16_t` `PackCreateProfileOut::ExtErrorCode`

8.612.2.2 `uint8_t` `PackCreateProfileOut::ProfileIndex`

8.612.2.3 `uint8_t` `PackCreateProfileOut::ProfileType`

## 8.613 packgetDyingGaspCfg Struct Reference

### Data Fields

- `uint8_t *` [pDestSMSNum](#)
- `uint8_t *` [pDestSMSCContent](#)

### 8.613.1 Detailed Description

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

## 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>

## 8.613.2 Field Documentation

8.613.2.1 uint8\_t\* packgetDyingGaspCfg::pDestSMSContent

8.613.2.2 uint8\_t\* packgetDyingGaspCfg::pDestSMSNum

## 8.614 packgetDyingGaspStatistics Struct Reference

## Data Fields

- uint32\_t \* [pTimeStamp](#)
- uint8\_t \* [pSMSAttemptedFlag](#)

## 8.614.1 Detailed Description

This structure contains Get Dying GASP Statistics

## Parameters

<i>pTimeStamp</i> [OUT]	<ul style="list-style-type: none"> <li>Timestamp of the last time power loss was detected and Dying Gasp feature was triggered.</li> <li>UTC time in seconds since Jan 06, 1980 (GPS Epoch).</li> </ul>
<i>pSMSAttemptedFlag</i> [OUT]	<ul style="list-style-type: none"> <li>Indicates whether device attempted to send SMS in the last power loss event.</li> <li>0 - SMS not attempted</li> <li>1 - SMS attempted</li> <li>This only indicates device sent the SMS but does not guarantee network delivery.</li> </ul>

## 8.614.2 Field Documentation

8.614.2.1 uint8\_t\* packgetDyingGaspStatistics::pSMSAttemptedFlag

8.614.2.2 uint32\_t\* packgetDyingGaspStatistics::pTimeStamp

## 8.615 PackSwiAvmsSetSettingsAPNInfo Struct Reference

## Data Fields

- uint8\_t [bAPNLength](#)
- uint8\_t [szAPN](#) [49]

- uint8\_t [bUnameLength](#)
- uint8\_t [szUname](#) [29]
- uint8\_t [bPWDLenght](#)
- uint8\_t [szPWD](#) [29]

### 8.615.1 Detailed Description

This structure contains pack set settings APN information.

#### Parameters

<i>bAPNLength</i>	APN Length.
<i>szAPN</i>	APN String.
<i>bUnameLength</i>	Username Length.
<i>szUname</i>	User Name String.
<i>bPWDLenght</i>	Password Length.
<i>szPWD</i>	Password String.

### 8.615.2 Field Documentation

8.615.2.1 uint8\_t PackSwiAvmsSetSettingsAPNInfo::bAPNLength

8.615.2.2 uint8\_t PackSwiAvmsSetSettingsAPNInfo::bPWDLenght

8.615.2.3 uint8\_t PackSwiAvmsSetSettingsAPNInfo::bUnameLength

8.615.2.4 uint8\_t PackSwiAvmsSetSettingsAPNInfo::szAPN[49]

8.615.2.5 uint8\_t PackSwiAvmsSetSettingsAPNInfo::szPWD[29]

8.615.2.6 uint8\_t PackSwiAvmsSetSettingsAPNInfo::szUname[29]

## 8.616 PackSwiAvmsSetSettingsConnectionRetryTimers Struct Reference

### Data Fields

- uint16\_t [Timers](#) [8]

### 8.616.1 Detailed Description

This structure contains the SLQSAVMSSetSettings Polling timer to connect to AVMS server.

#### Parameters

<i>Timers[0]</i>	- Timer 1. <ul style="list-style-type: none"><li>• 0-20160 (min)</li><li>• 0:retry disabled</li></ul>
<i>Timers[1]</i>	- Timer 2. <ul style="list-style-type: none"><li>• 1-20160 (min)</li></ul>
<i>Timers[2]</i>	- Timer 3. <ul style="list-style-type: none"><li>• 1-20160 (min)</li></ul>

<i>Timers[3]</i>	- Timer 4. • 1-20160 (min)
<i>Timers[4]</i>	- Timer 5. • 1-20160 (min)
<i>Timers[5]</i>	- Timer 6. • 1-20160 (min)
<i>Timers[6]</i>	- Timer 7. • 1-20160 (min)
<i>Timers[7]</i>	- Timer 8. • 1-20160 (min)

## 8.616.2 Field Documentation

8.616.2.1 uint16\_t PackSwiAvmsSetSettingsConnectionRetryTimers::Timers[8]

## 8.617 PackSwiAvmsSetSettingsPeriodInfo Struct Reference

### Data Fields

- uint32\_t [ulMin](#)
- uint32\_t [ulMax](#)

### 8.617.1 Detailed Description

This structure contain pack set settings period information.

#### Parameters

<i>ulMin</i>	Min period in seconds.
<i>ulMax</i>	Max period in seconds.

## 8.617.2 Field Documentation

8.617.2.1 uint32\_t PackSwiAvmsSetSettingsPeriodInfo::ulMax

8.617.2.2 uint32\_t PackSwiAvmsSetSettingsPeriodInfo::ulMin

## 8.618 PackSwiAVMSSettingsAPNInfo Struct Reference

### Data Fields

- uint8\_t [bAPNLength](#)
- uint8\_t \* [pAPN](#)
- uint8\_t [bUnameLength](#)
- uint8\_t \* [pUname](#)
- uint8\_t [bPWDLength](#)

- uint8\_t \* [pPWD](#)

### 8.618.1 Detailed Description

This structure contains the SLQSAVMSGetSettings APN Info to connect to AVMS server.

#### Parameters

<i>bAPNLength</i>	- APN Length(Max 49).
<i>szAPN</i>	- APN.
<i>bUnameLength</i>	- User Name Length (Max 29).
<i>szUname</i>	- User Name.
<i>bPWDLength</i>	- Password Length (Max 29).
<i>szPWD</i>	- Password.

### 8.618.2 Field Documentation

8.618.2.1 uint8\_t PackSwiAVMSSettingsAPNInfo::bAPNLength

8.618.2.2 uint8\_t PackSwiAVMSSettingsAPNInfo::bPWDLength

8.618.2.3 uint8\_t PackSwiAVMSSettingsAPNInfo::bUnameLength

8.618.2.4 uint8\_t\* PackSwiAVMSSettingsAPNInfo::pAPN

8.618.2.5 uint8\_t\* PackSwiAVMSSettingsAPNInfo::pPWD

8.618.2.6 uint8\_t\* PackSwiAVMSSettingsAPNInfo::pUname

## 8.619 PackSwiAVMSSettingsConnectionRetryTimers Struct Reference

### Data Fields

- uint16\_t [Timers](#) [8]

### 8.619.1 Detailed Description

This structure contains the SLQSAVMSGetSettings Polling timer to connect to AVMS server.

#### Parameters

<i>Timers[0]</i>	- Timer 1. <ul style="list-style-type: none"> <li>• 0-20160 (min)</li> <li>• 0:retry disabled</li> </ul>
<i>Timers[1]</i>	- Timer 2. <ul style="list-style-type: none"> <li>• 1-20160 (min)</li> </ul>
<i>Timers[2]</i>	- Timer 3. <ul style="list-style-type: none"> <li>• 1-20160 (min)</li> </ul>
<i>Timers[3]</i>	- Timer 4. <ul style="list-style-type: none"> <li>• 1-20160 (min)</li> </ul>

<i>Timers[4]</i>	- Timer 5. • 1-20160 (min)
<i>Timers[5]</i>	- Timer 6. • 1-20160 (min)
<i>Timers[6]</i>	- Timer 7. • 1-20160 (min)
<i>Timers[7]</i>	- Timer 8. • 1-20160 (min)

## 8.619.2 Field Documentation

8.619.2.1 `uint16_t PackSwiAVMSSettingsConnectionRetryTimers::Timers[8]`

## 8.620 PackSwiAVMSSettingsPeriodsInfo Struct Reference

### Data Fields

- `uint32_t` [min](#)
- `uint32_t` [max](#)

### 8.620.1 Detailed Description

This structure contains the SLQSAVMSSettings Min/Max Period of an Observation.

#### Parameters

<i>min</i>	- Min period in seconds.
<i>max</i>	- Max period in seconds.

## 8.620.2 Field Documentation

8.620.2.1 `uint32_t PackSwiAVMSSettingsPeriodsInfo::max`

8.620.2.2 `uint32_t PackSwiAVMSSettingsPeriodsInfo::min`

## 8.621 qmiSmsMessageList Struct Reference

### Data Fields

- `uint32_t` [messageIndex](#)
- `uint32_t` [messageTag](#)

### 8.621.1 Detailed Description

This structure contains SMS message list.

## 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>

## 8.621.2 Field Documentation

8.621.2.1 uint32\_t qmSmsMessageList::messageIndex

8.621.2.2 uint32\_t qmSmsMessageList::messageTag

## 8.622 qmWSDDataBearerTechnology Struct Reference

## Data Fields

- uint8\_t [currentNetwork](#)
- uint32\_t [ratMask](#)
- uint32\_t [soMask](#)

## 8.622.1 Detailed Description

Structure to hold the current data bearer technology values

## Parameters

<i>currentNetwork[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> <li>– 0xff - Invalid data.</li> </ul> </li> </ul>
<i>ratMask[OUT]</i>	<a href="#">RAT Mask</a>
<i>soMask[OUT]</i>	<a href="#">SO Mask</a>

## 8.622.2 Field Documentation

8.622.2.1 uint8\_t qmWSDDataBearerTechnology::currentNetwork

8.622.2.2 uint32\_t qmWSDDataBearerTechnology::ratMask

8.622.2.3 uint32\_t qmWSDDataBearerTechnology::soMask

## 8.623 qmTlvResult Struct Reference

## Data Fields

- [swi\\_uint256\\_t TlvPresenceMask](#)
- [qmulong TlvResultCode](#)

- [qmuint16 DeviceResult](#)
- [qmuint16 DeviceError](#)

### 8.623.1 Field Documentation

8.623.1.1 [qmuint16 qmTlvResult::DeviceError](#)

8.623.1.2 [qmuint16 qmTlvResult::DeviceResult](#)

8.623.1.3 [swi\\_uint256\\_t qmTlvResult::TlvPresenceMask](#)

8.623.1.4 [qmulong qmTlvResult::TlvResultCode](#)

## 8.624 qos\_BindDataPortMuxID\_t Struct Reference

### Data Fields

- [uint8\\_t MuxID](#)

### 8.624.1 Detailed Description

Structure that contains the request Binds a control point to a data port Mux ID data.

#### Parameters

<i>MuxID</i>	Mux ID: <ul style="list-style-type: none"> <li>• 0x80-0x8F : valid value</li> </ul>
--------------	---

### 8.624.2 Field Documentation

8.624.2.1 [uint8\\_t qos\\_BindDataPortMuxID\\_t::MuxID](#)

## 8.625 qos\_BindDataPortPeripheralEndPointID\_t Struct Reference

### Data Fields

- [uint32\\_t EndPointerType](#)
- [uint32\\_t lfaceID](#)

### 8.625.1 Detailed Description

Structure that contains the request Binds a control point to a data port Peripheral End Point Type data.

#### Parameters

<i>EndPointType</i>	Peripheral end point type. Values: <ul style="list-style-type: none"> <li>• 0 - Reserved</li> <li>• 1 - HSIC</li> <li>• 2 - HSUSB</li> <li>• 3 - PCIE</li> <li>• 4 - Embedded</li> </ul>
---------------------	--



<i>IfaceID</i>	Peripheral interface number.
----------------	------------------------------

## 8.625.2 Field Documentation

8.625.2.1 uint32\_t qos\_BindDataPortPeripheralEndPointID\_t::EndPointType

8.625.2.2 uint32\_t qos\_BindDataPortPeripheralEndPointID\_t::IfaceID

## 8.626 qos\_BindDataPortSIODataPort\_t Struct Reference

### Data Fields

- uint16\_t [SIODataPort](#)

### 8.626.1 Detailed Description

Structure that contains the request Binds a control point to a data port SIO Data Port data.

#### Parameters

<i>u16SIODataPort</i>	SIO Data Port to which the client binds
-----------------------	---

## 8.626.2 Field Documentation

8.626.2.1 uint16\_t qos\_BindDataPortSIODataPort\_t::SIODataPort

## 8.627 RFBandInfoElements Struct Reference

### Data Fields

- uint8\_t [radioInterface](#)
- uint16\_t [activeBandClass](#)
- uint16\_t [activeChannel](#)

### 8.627.1 Detailed Description

This structure contains the RFBandInfo response parameters.

#### Parameters

<i>radioInterface</i>	<ul style="list-style-type: none"> <li>• Radio interface technology <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Radio Interface</li> </ul> </li> </ul>
<i>activeBandClass</i>	<ul style="list-style-type: none"> <li>• Active Band Class <ul style="list-style-type: none"> <li>– See <a href="#">Tables</a> for Band Classes</li> </ul> </li> </ul>
<i>activeChannel</i>	<ul style="list-style-type: none"> <li>• Active channel (0 if channel is not relevant to the reported technology)</li> </ul>

## 8.627.2 Field Documentation

8.627.2.1 uint16\_t RFBandInfoElements::activeBandClass

8.627.2.2 uint16\_t RFBandInfoElements::activeChannel

8.627.2.3 uint8\_t RFBandInfoElements::radioInterface

## 8.628 rmTrasnferStaticsReq Struct Reference

### Data Fields

- uint8\_t [bResetStatistics](#)
- uint32\_t [ulMask](#)

### 8.628.1 Detailed Description

RM Transfer Statistics Structure

#### Parameters

<i>bResetStatistics</i>	<ul style="list-style-type: none"> <li>• Reset Statistics</li> <li>• Values: <ul style="list-style-type: none"> <li>• 0 - Not Reset</li> <li>• Other - Reset</li> </ul> </li> </ul>
<i>ulMask</i>	<ul style="list-style-type: none"> <li>• Enable/Disable RM Transfer Satatistics Indiscation Mask</li> <li>• Bit 0: Tx Packet Ok</li> <li>• Bit 1: Rx Packet Ok</li> <li>• Bit 2: Tx Bytes Ok</li> <li>• Bit 3: Rx Bytes Ok</li> <li>• Bit 4: Tx Packets Dropped</li> <li>• Bit 5: Rx Packets Dropped</li> <li>• Value: -0 - Disable -1 - Enable</li> </ul>

## 8.628.2 Field Documentation

8.628.2.1 uint8\_t rmTrasnferStaticsReq::bResetStatistics

8.628.2.2 uint32\_t rmTrasnferStaticsReq::ulMask

## 8.629 sensorData\_t Struct Reference

### Data Fields

- uint32\_t [timeOfFirstSample](#)
- uint8\_t [flags](#)
- uint8\_t [sensorDataLen](#)
- uint16\_t [timeOffset](#) [64]

- uint32\_t [xAxis](#) [64]
- uint32\_t [yAxis](#) [64]
- uint32\_t [zAxis](#) [64]

### 8.629.1 Detailed Description

This structure specifies information regarding the 3-Axis Sensor Data. Please check has\_<Param\_Name> field for presence of optional parameters

#### 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>

### 8.629.2 Field Documentation

8.629.2.1 `uint8_t sensorData_t::flags`

8.629.2.2 `uint8_t sensorData_t::sensorDataLen`

8.629.2.3 `uint32_t sensorData_t::timeOfFirstSample`

8.629.2.4 `uint16_t sensorData_t::timeOffset[64]`

8.629.2.5 `uint32_t sensorData_t::xAxis[64]`

8.629.2.6 `uint32_t sensorData_t::yAxis[64]`

8.629.2.7 `uint32_t sensorData_t::zAxis[64]`

## 8.630 `slot_t` Struct Reference

### Data Fields

- `uint32_t uPhyCardStatus`
- `uint32_t uPhySlotStatus`
- `uint8_t bLogicalSlot`
- `uint8_t bICCIDLength`
- `uint8_t bICCID [255]`

### 8.630.1 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>

## 8.630.2 Field Documentation

8.630.2.1 uint8\_t slot\_t::bICCID[255]

8.630.2.2 uint8\_t slot\_t::bICCIDLength

8.630.2.3 uint8\_t slot\_t::bLogicalSlot

8.630.2.4 uint32\_t slot\_t::uPhyCardStatus

8.630.2.5 uint32\_t slot\_t::uPhySlotStatus

## 8.631 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]

### 8.631.1 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>

## 8.631.2 Field Documentation

8.631.2.1 `appStats slotInf::AppStatus[10]`

8.631.2.2 `uint8_t slotInf::cardState`

8.631.2.3 `uint8_t slotInf::errorState`

8.631.2.4 `uint8_t slotInf::numApp`

8.631.2.5 `uint8_t slotInf::upinRetries`

8.631.2.6 `uint8_t slotInf::upinState`

8.631.2.7 `uint8_t slotInf::upukRetries`

## 8.632 slots\_t Struct Reference

### Data Fields

- [slot\\_t uimSlotStatus](#) [255]

### 8.632.1 Detailed Description

This structure contains slots status.

## Parameters

<i>uimSlotStatus</i>	<ul style="list-style-type: none"> <li>• See <a href="#">slot_t</a></li> </ul>
----------------------	--

## 8.632.2 Field Documentation

8.632.2.1 slot\_t slots\_t::uimSlotStatus[255]

## 8.633 sms\_BroadcastConfig Struct Reference

## Data Fields

- uint16\_t [fromServiceId](#)
- uint16\_t [toServiceId](#)
- uint8\_t [selected](#)

## 8.633.1 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>

## 8.633.2 Field Documentation

8.633.2.1 uint16\_t sms\_BroadcastConfig::fromServiceId

8.633.2.2 uint8\_t sms\_BroadcastConfig::selected

8.633.2.3 uint16\_t sms\_BroadcastConfig::toServiceId

## 8.634 sms\_CDMABroadcastConfig Struct Reference

## Data Fields

- uint16\_t [serviceCategory](#)
- uint16\_t [language](#)
- uint8\_t [selected](#)

### 8.634.1 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>

### 8.634.2 Field Documentation

8.634.2.1 uint16\_t sms\_CDMABroadcastConfig::language

8.634.2.2 uint8\_t sms\_CDMABroadcastConfig::selected

8.634.2.3 uint16\_t sms\_CDMABroadcastConfig::serviceCategory

## 8.635 sms\_getIndicationReg Struct Reference

#### Data Fields

- uint8\_t \* [pRegTransLayerInfoEvt](#)
- uint8\_t \* [pRegTransNWRegInfoEvt](#)
- uint8\_t \* [pRegCallStatInfoEvt](#)

### 8.635.1 Detailed Description

This structure contains Get Indication Register Response parameters

#### Parameters

<i>pRegTransLayerInfoEvt</i>	<ul style="list-style-type: none"> <li>-</li> <li>• Optional 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> </ul>
<i>pRegTransNWRegInfoEvt</i>	<ul style="list-style-type: none"> <li>-</li> <li>• Optional 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> </ul>



<i>pRegCallStatInfoEvt</i>	- <ul style="list-style-type: none"> <li>Optional 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> </ul>
----------------------------	---

## 8.635.2 Field Documentation

8.635.2.1 `uint8_t* sms_getIndicationReg::pRegCallStatInfoEvt`

8.635.2.2 `uint8_t* sms_getIndicationReg::pRegTransLayerInfoEvt`

8.635.2.3 `uint8_t* sms_getIndicationReg::pRegTransNWRegInfoEvt`

## 8.636 sms\_getMsgWaitingInfo Struct Reference

### Data Fields

- `uint8_t numInstances`
- `sms_messageWaitingInfoContent msgWaitInfo` [0xFF]

### 8.636.1 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 <code>sms_messageWaitingInfoContent</code></li> </ul>
<i>pMsgWaitInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <code>sms_messageWaitingInfoContent</code>. <ul style="list-style-type: none"> <li>See <code>sms_messageWaitingInfoContent</code> for more information.</li> </ul> </li> </ul>

## 8.636.2 Field Documentation

8.636.2.1 `sms_messageWaitingInfoContent sms_getMsgWaitingInfo::msgWaitInfo`[0xFF]

8.636.2.2 `uint8_t sms_getMsgWaitingInfo::numInstances`

## 8.637 sms\_getTransLayerInfo Struct Reference

### Data Fields

- `uint8_t * pRegInd`
- `sms_transLayerInfo * pTransLayerInfo`

### 8.637.1 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>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 <a href="#">sms_transLayerInfo</a>. <ul style="list-style-type: none"> <li>Optional parameter</li> <li>See <a href="#">sms_transLayerInfo</a> for more information</li> </ul> </li> <li>returns a default value 0xFF for parameter values if no response is received from the device.</li> </ul>

### 8.637.2 Field Documentation

8.637.2.1 `uint8_t* sms_getTransLayerInfo::pRegInd`

8.637.2.2 `sms_transLayerInfo* sms_getTransLayerInfo::pTransLayerInfo`

## 8.638 sms\_getTransNWRegInfo Struct Reference

### Data Fields

- `uint8_t * pRegStatus`

### 8.638.1 Detailed Description

This structure contains transport network registration info parameter

#### Parameters

<i>pRegStatus</i>	- <ul style="list-style-type: none"> <li>Optional 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>returns a default value 0xFF if no response is received from the device.</li> </ul>
-------------------	---

### 8.638.2 Field Documentation

8.638.2.1 uint8\_t\* sms\_getTransNWRegInfo::pRegStatus

## 8.639 sms\_maxStorageSizeReq Struct Reference

### Data Fields

- uint8\_t [storageType](#)
- uint8\_t \* [pMessageMode](#)

### 8.639.1 Detailed Description

This structure contains get store max size request 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

### 8.639.2 Field Documentation

8.639.2.1 uint8\_t\* sms\_maxStorageSizeReq::pMessageMode

8.639.2.2 uint8\_t sms\_maxStorageSizeReq::storageType

## 8.640 sms\_maxStorageSizeResp Struct Reference

### Data Fields

- uint32\_t [maxStorageSize](#)
- uint32\_t [freeSlots](#)

### 8.640.1 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>returns a default value 0xFFFFFFFF for parameter values if no response is received from the device.</li> </ul>
------------------	---

### 8.640.2 Field Documentation

8.640.2.1 uint32\_t sms\_maxStorageSizeResp::freeSlots

8.640.2.2 uint32\_t sms\_maxStorageSizeResp::maxStorageSize

## 8.641 sms\_messageWaitingInfoContent Struct Reference

### Data Fields

- uint8\_t [msgType](#)
- uint8\_t [activeInd](#)
- uint8\_t [msgCount](#)

### 8.641.1 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_VOICMAIL - 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>

### 8.641.2 Field Documentation

8.641.2.1 uint8\_t sms\_messageWaitingInfoContent::activeInd

8.641.2.2 uint8\_t sms\_messageWaitingInfoContent::msgCount

8.641.2.3 uint8\_t sms\_messageWaitingInfoContent::msgType

## 8.642 sms\_msgProtocolResp Struct Reference

### Data Fields

- uint8\_t [msgProtocol](#)

### 8.642.1 Detailed Description

This structure contains get message protocol response parameters

#### Parameters

<i>msgProtocol</i>	- <ul style="list-style-type: none"> <li>• Message Protocol</li> <li>• Values:             <ul style="list-style-type: none"> <li>– 0x00 - MESSAGE_PROTOCOL_CDMA</li> <li>– 0x01 - MESSAGE_PROTOCOL_WCDMA</li> </ul> </li> </ul>
--------------------	--

### 8.642.2 Field Documentation

8.642.2.1 uint8\_t sms\_msgProtocolResp::msgProtocol

## 8.643 sms\_qaQmi3GPP2BroadcastCfgInfo Struct Reference

### Data Fields

- uint8\_t [activated\\_ind](#)
- uint16\_t [num\\_instances](#)
- [sms\\_CDMABroadcastConfig](#) [CDMABroadcastConfig](#) [0x05]

### 8.643.1 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 CDMABroadcastConfig structure array.</li> <li>• Further defined by the structure <a href="#">sms_CDMABroadcastConfig</a></li> </ul>

### 8.643.2 Field Documentation

8.643.2.1 `uint8_t sms_qaQmi3GPP2BroadcastCfgInfo::activated_ind`

8.643.2.2 `sms_CDMABroadcastConfig sms_qaQmi3GPP2BroadcastCfgInfo::CDMABroadcastConfig[0x05]`

8.643.2.3 `uint16_t sms_qaQmi3GPP2BroadcastCfgInfo::num_instances`

## 8.644 sms\_qaQmi3GPPBroadcastCfgInfo Struct Reference

### Data Fields

- `uint8_t activated_ind`
- `uint16_t num_instances`
- `sms_BroadcastConfig broadcastConfig` [0x05]

### 8.644.1 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 BroadcastConfig structure array.</li> <li>• Further defined by the structure <a href="#">sms_BroadcastConfig</a></li> </ul>

### 8.644.2 Field Documentation

8.644.2.1 `uint8_t sms_qaQmi3GPPBroadcastCfgInfo::activated_ind`

8.644.2.2 `sms_BroadcastConfig sms_qaQmi3GPPBroadcastCfgInfo::broadcastConfig[0x05]`

8.644.2.3 `uint16_t sms_qaQmi3GPPBroadcastCfgInfo::num_instances`

## 8.645 sms\_routeEntry Struct Reference

### Data Fields

- `uint8_t messageType`
- `uint8_t messageClass`

- uint8\_t [routeStorage](#)
- uint8\_t [receiptAction](#)

### 8.645.1 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>

### 8.645.2 Field Documentation

8.645.2.1 uint8\_t sms\_routeEntry::messageClass

8.645.2.2 uint8\_t sms\_routeEntry::messageType

8.645.2.3 uint8\_t sms\_routeEntry::receiptAction

8.645.2.4 uint8\_t sms\_routeEntry::routeStorage

## 8.646 sms\_sendAsyncsmsParams Struct Reference

## Data Fields

- uint32\_t [messageFormat](#)
- uint32\_t [messageSize](#)
- uint8\_t \* [pMessage](#)
- uint8\_t \* [pForceOnDC](#)
- uint8\_t \* [pServiceOption](#)
- uint8\_t \* [pFollowOnDC](#)
- uint8\_t \* [pLinktimer](#)
- uint8\_t \* [pSmsOnIms](#)
- uint8\_t \* [pRetryMessage](#)
- uint32\_t \* [pRetryMessageld](#)
- uint32\_t \* [pUserData](#)

### 8.646.1 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>

## 8.646.2 Field Documentation

8.646.2.1 `uint32_t sms_sendAsyncsmsParams::messageFormat`

8.646.2.2 `uint32_t sms_sendAsyncsmsParams::messageSize`

8.646.2.3 `uint8_t* sms_sendAsyncsmsParams::pFollowOnDC`

8.646.2.4 `uint8_t* sms_sendAsyncsmsParams::pForceOnDC`

8.646.2.5 `uint8_t* sms_sendAsyncsmsParams::pLinktimer`

8.646.2.6 `uint8_t* sms_sendAsyncsmsParams::pMessage`

8.646.2.7 `uint8_t* sms_sendAsyncsmsParams::pRetryMessage`

8.646.2.8 `uint32_t* sms_sendAsyncsmsParams::pRetryMessageId`

8.646.2.9 `uint8_t* sms_sendAsyncsmsParams::pServiceOption`

8.646.2.10 `uint8_t* sms_sendAsyncsmsParams::pSmsOnIms`

8.646.2.11 `uint32_t* sms_sendAsyncsmsParams::pUserData`

## 8.647 sms\_setIndicationReg Struct Reference

### Data Fields

- uint8\_t \* [pRegTransLayerInfoEvt](#)
- uint8\_t \* [pRegTransNWRegInfoEvt](#)
- uint8\_t \* [pRegCallStatInfoEvt](#)

### 8.647.1 Detailed Description

This structure contains Indication Register request parameters

#### Parameters

<i>pRegTransLayerInfoEvt</i>	- <ul style="list-style-type: none"> <li>• Optional 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 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 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>

### 8.647.2 Field Documentation

8.647.2.1 uint8\_t\* sms\_setIndicationReg::pRegCallStatInfoEvt

8.647.2.2 uint8\_t\* sms\_setIndicationReg::pRegTransLayerInfoEvt

8.647.2.3 uint8\_t\* sms\_setIndicationReg::pRegTransNWRegInfoEvt

## 8.648 sms\_setRoutesReq Struct Reference

### Data Fields

- uint16\_t [numOfRoutes](#)

- [sms\\_routeEntry](#) routeList [0x0A]
- uint8\_t \* [pTransferStatusReport](#)

### 8.648.1 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="#">sms_routeEntry</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>

### 8.648.2 Field Documentation

8.648.2.1 uint16\_t sms\_setRoutesReq::numOfRoutes

8.648.2.2 uint8\_t\* sms\_setRoutesReq::pTransferStatusReport

8.648.2.3 sms\_routeEntry sms\_setRoutesReq::routeList[0x0A]

## 8.649 sms\_transLayerInfo Struct Reference

### Data Fields

- uint8\_t [TransType](#)
- uint8\_t [TransCap](#)

### 8.649.1 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>

### 8.649.2 Field Documentation

8.649.2.1 `uint8_t sms_transLayerInfo::TransCap`

8.649.2.2 `uint8_t sms_transLayerInfo::TransType`

## 8.650 sMSCAddressInfo Struct Reference

### Data Fields

- `uint8_t length`
- `uint8_t data` [256]

### 8.650.1 Detailed Description

This structure contains 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>

### 8.650.2 Field Documentation

8.650.2.1 `uint8_t sMSCAddressInfo::data`[256]

8.650.2.2 `uint8_t sMSCAddressInfo::length`

## 8.651 sMSCAddressTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `sMSCAddressInfo SMSCInfo`

### 8.651.1 Detailed Description

This structure contains SMSC TLV information

#### 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 <code>sMSCAddressInfo</code> for more information</li> </ul>

### 8.651.2 Field Documentation

8.651.2.1 sMSCAddressInfo sMSCAddressTlv::SMSCInfo

8.651.2.2 uint8\_t sMSCAddressTlv::TlvPresent

## 8.652 sMSEtwsMessageInfo Struct Reference

### Data Fields

- uint8\_t [notificationType](#)
- uint16\_t [length](#)
- uint8\_t [data](#) [1254]

### 8.652.1 Detailed Description

This structure contains ETWS information

#### 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>

### 8.652.2 Field Documentation

8.652.2.1 uint8\_t sMSEtwsMessageInfo::data[1254]

8.652.2.2 uint16\_t sMSEtwsMessageInfo::length

8.652.2.3 uint8\_t sMSEtwsMessageInfo::notificationType

## 8.653 sMSEtwsMessageTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- [sMSEtwsMessageInfo](#) [EtwsMessageInfo](#)

### 8.653.1 Detailed Description

This structure contains ETWS TLV information

#### 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="#">sMSEtwsMessageTlv::EtwsMessageInfo</a> for more information</li> </ul>
-------------------------	---

### 8.653.2 Field Documentation

8.653.2.1 `sMSEtwsMessageInfo sMSEtwsMessageTlv::EtwsMessageInfo`

8.653.2.2 `uint8_t sMSEtwsMessageTlv::TlvPresent`

## 8.654 sMSEtwsPlmnInfo Struct Reference

### Data Fields

- `uint16_t` [mobileCountryCode](#)
- `uint16_t` [mobileNetworkCode](#)

### 8.654.1 Detailed Description

This structure contains ETWS PLMN information

#### 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>

### 8.654.2 Field Documentation

8.654.2.1 `uint16_t sMSEtwsPlmnInfo::mobileCountryCode`

8.654.2.2 `uint16_t sMSEtwsPlmnInfo::mobileNetworkCode`

## 8.655 sMSMessageModeInfo Struct Reference

### Data Fields

- `uint8_t` [messageMode](#)

### 8.655.1 Detailed Description

This structure contains message mode information.

## Parameters

<i>messageMode</i>	<ul style="list-style-type: none"> <li>• Message Mode <ul style="list-style-type: none"> <li>– 0x00 - CDMA</li> <li>– 0x01 - GW</li> </ul> </li> </ul>
--------------------	--

## 8.655.2 Field Documentation

8.655.2.1 uint8\_t sMSMessageModelInfo::messageMode

## 8.656 sMSMTMessageInfo Struct Reference

## Data Fields

- uint32\_t [storageType](#)
- uint32\_t [messageIndex](#)

## 8.656.1 Detailed Description

This structure contains MT message information.

## Parameters

<i>storageType</i>	memory storage 0x00-UIIM 0x01-NV
<i>messageIndex</i>	MT Message index

## 8.656.2 Field Documentation

8.656.2.1 uint32\_t sMSMTMessageInfo::messageIndex

8.656.2.2 uint32\_t sMSMTMessageInfo::storageType

## 8.657 sMSOnIMSInfo Struct Reference

## Data Fields

- uint8\_t [smsOnIMS](#)

## 8.657.1 Detailed Description

This structure contains SMS on IMS information

## Parameters

<i>smsOnIMS</i>	SMS on IMS
-----------------	------------

## 8.657.2 Field Documentation

8.657.2.1 uint8\_t sMSOnIMSInfo::smsOnIMS

## 8.658 sMSOnIMSTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- [sMSOnIMSInfo](#) [IMSInfo](#)

### 8.658.1 Detailed Description

This structure contains SMS on IMS TLV information.

#### 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>

### 8.658.2 Field Documentation

8.658.2.1 [sMSOnIMSInfo](#) [sMSOnIMSTlv::IMSInfo](#)

8.658.2.2 [uint8\\_t](#) [sMSOnIMSTlv::TlvPresent](#)

## 8.659 sMSTransferRouteMTMessageInfo Struct Reference

### Data Fields

- [uint8\\_t](#) [ackIndicator](#)
- [uint32\\_t](#) [transactionID](#)
- [uint8\\_t](#) [format](#)
- [uint16\\_t](#) [length](#)
- [uint8\\_t](#) [data](#) [256]

### 8.659.1 Detailed Description

This structure contains Transfer route MT message information.

#### 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>

### 8.659.2 Field Documentation

8.659.2.1 uint8\_t sMSTransferRouteMTMessageInfo::ackIndicator

8.659.2.2 uint8\_t sMSTransferRouteMTMessageInfo::data[256]

8.659.2.3 uint8\_t sMSTransferRouteMTMessageInfo::format

8.659.2.4 uint16\_t sMSTransferRouteMTMessageInfo::length

8.659.2.5 uint32\_t sMSTransferRouteMTMessageInfo::transactionID

## 8.660 swi\_uint256\_t Struct Reference

### Data Fields

- uint16\_t [word](#) [16]

### 8.660.1 Detailed Description

SWI 256 bit data type

### 8.660.2 Field Documentation

8.660.2.1 uint16\_t swi\_uint256\_t::word[16]

## 8.661 swiaudio\_PCMparams Struct Reference

### Data Fields

- uint8\_t [iFaceTabLen](#)
- uint8\_t [iFaceTab](#) [255]

### 8.661.1 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>
-----------------	---

### 8.661.2 Field Documentation

8.661.2.1 `uint8_t swiaudio_PCMparams::iFaceTab[255]`

8.661.2.2 `uint8_t swiaudio_PCMparams::iFaceTabLen`

## 8.662 swidms\_ehrpdMTUSizeTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint16_t ehrpdMTUSize`

### 8.662.1 Detailed Description

This structure contains the parameters for EHRPD MTU Size.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>
<i>ehrpMTUSize</i>	EHRPD size <ul style="list-style-type: none"> <li>Size of EHRPD MTU</li> </ul>

### 8.662.2 Field Documentation

8.662.2.1 `uint16_t swidms_ehrpdMTUSizeTlv::ehrpMTUSize`

8.662.2.2 `uint8_t swidms_ehrpdMTUSizeTlv::TlvPresent`

## 8.663 swidms\_hrpMTUSizeTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint16_t hrpdMTUSize`

### 8.663.1 Detailed Description

This structure contains the parameters for HRPD MTU Size.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Tlv Present.</li></ul>
<i>hrpdMTUSize</i>	HRPD size <ul style="list-style-type: none"><li>• Size of HRPD MTU</li></ul>

## 8.663.2 Field Documentation

8.663.2.1 uint16\_t swidms\_hrpdmTUSizeTlv::hrpdMTUSize

8.663.2.2 uint8\_t swidms\_hrpdmTUSizeTlv::TlvPresent

## 8.664 swidms\_ifaceCfgTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [CurrentCfgType](#)
- uint32\_t [CfgValue](#)

## 8.664.1 Detailed Description

This structure contains the parameters for host usb interface composition

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Tlv Present.</li></ul>
<i>CurrentCfgType</i>	<ul style="list-style-type: none"><li>• Current configure type, see values below:<ul style="list-style-type: none"><li>– 1 - MBIM</li><li>– 2 - USBIF + PCIE</li><li>– 3 - RMNET</li><li>– 4 - RNDIS</li><li>– 5 - PCIE only</li></ul></li></ul>

<i>CfgValue</i>	<ul style="list-style-type: none"> <li>• Host composition bit mask, see supported USB interface bitmasks, not supported by QMI object setting for 9x50 modules <ul style="list-style-type: none"> <li>– 0x00000001 - DIAG interface</li> <li>– 0x00000002 - ADB interface</li> <li>– 0x00000004 - NMEA interface</li> <li>– 0x00000008 - MODEM interface</li> <li>– 0x00000010 - RESERVED5</li> <li>– 0x00000020 - RESERVED6</li> <li>– 0x00000040 - RESERVED7</li> <li>– 0x00000080 - RESERVED8</li> <li>– 0x00000100 - RMENT0 interface</li> <li>– 0x00000200 - RESERVED10</li> <li>– 0x00000400 - RMENT1 interface</li> <li>– 0x00000800 - RESERVED12</li> <li>– 0x00001000 - MBIM interface</li> <li>– 0x00002000 - RESERVED14</li> <li>– 0x00004000 - RNDIS interface</li> <li>– 0x00008000 - RESERVED16</li> <li>– 0x00010000 - AUDIO interface</li> <li>– 0x00020000 - RESERVED18</li> <li>– 0x00080000 - ECM interface</li> <li>– 0x00100000 - RESERVED21</li> <li>– 0x00200000 - RESERVED22</li> <li>– 0xFFC00000 - RESERVED</li> </ul> </li> </ul>
-----------------	--

## 8.664.2 Field Documentation

8.664.2.1 `uint32_t swidms_ifaceCfgTlv::CfgValue`

8.664.2.2 `uint32_t swidms_ifaceCfgTlv::CurrentCfgType`

8.664.2.3 `uint8_t swidms_ifaceCfgTlv::TlvPresent`

## 8.665 swidms\_mtuSize3gppTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint16_t MTUSize3gpp`

### 8.665.1 Detailed Description

This structure contains the parameters for 3GPP MTU Size.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
-------------------	--

<i>MTUSize3gpp</i>	3GPP MTU size <ul style="list-style-type: none"> <li>Size of 3GPP MTU</li> </ul>
--------------------	--

### 8.665.2 Field Documentation

8.665.2.1 `uint16_t swidms_mtuSize3gppTlv::MTUSize3gpp`

8.665.2.2 `uint8_t swidms_mtuSize3gppTlv::TlvPresent`

## 8.666 swidms\_supportedIntBitmaskTlv Struct Reference

### Data Fields

- `uint8_t` [TlvPresent](#)
- `uint32_t` [ValidBitmasks](#)

### 8.666.1 Detailed Description

This structure contains the parameters for supported usb bitmasks

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>
<i>ValidBitmasks</i>	<ul style="list-style-type: none"> <li>for current configuration type, this is the supported bitmasks that the host can change.</li> </ul>

### 8.666.2 Field Documentation

8.666.2.1 `uint8_t swidms_supportedIntBitmaskTlv::TlvPresent`

8.666.2.2 `uint32_t swidms_supportedIntBitmaskTlv::ValidBitmasks`

## 8.667 swidms\_SwiDmsGetHWWatchdog Struct Reference

### Data Fields

- `uint32_t` [timeout](#)
- `uint32_t` [resetDelay](#)
- `uint8_t` [enable](#)
- `uint32_t` [count](#)

### 8.667.1 Detailed Description

This structure contains the parameters for hardware watchdog settings

## Parameters

<i>timeout</i>	<ul style="list-style-type: none"> <li>• timeout value for HW watchdog (unit in second)</li> </ul>
<i>resetDelay</i>	<ul style="list-style-type: none"> <li>• delay before reset after watchdog timeout (unit in second)</li> </ul>
<i>enable</i>	<ul style="list-style-type: none"> <li>• 0 to disable watchdog; 1 to enable watchdog</li> </ul>
<i>count</i>	<ul style="list-style-type: none"> <li>• once &lt;timeout&gt; has occurred, the &lt;count&gt; will be increased by 1 and the timer will be restarted automatically. This &lt;count&gt; indicates the number of renewals.</li> </ul>

## 8.667.2 Field Documentation

8.667.2.1 uint32\_t swidms\_SwiDmsGetHWWatchdog::count

8.667.2.2 uint8\_t swidms\_SwiDmsGetHWWatchdog::enable

8.667.2.3 uint32\_t swidms\_SwiDmsGetHWWatchdog::resetDelay

8.667.2.4 uint32\_t swidms\_SwiDmsGetHWWatchdog::timeout

## 8.668 swidms\_usbMTUSizeTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint16\_t [UsbMTUSize](#)

## 8.668.1 Detailed Description

This structure contains the parameters for USB MTU Size

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>
<i>UsbMTUSize</i>	USB MTU size <ul style="list-style-type: none"> <li>• Size of USB MTU</li> </ul>

## 8.668.2 Field Documentation

8.668.2.1 uint8\_t swidms\_usbMTUSizeTlv::TlvPresent

8.668.2.2 uint16\_t swidms\_usbMTUSizeTlv::UsbMTUSize

## 8.669 tdscdmaSigInfoExt Struct Reference

### Data Fields

- float [rssi](#)
- float [rscp](#)
- float [ecio](#)
- float [sinr](#)

### 8.669.1 Detailed Description

This structure contains the TDSCDMA Signal Strength Info Extended

#### Parameters

<i>rssi</i>	<ul style="list-style-type: none"> <li>• Measured RSSI in dB</li> </ul>
<i>rscp[Optional]</i>	<ul style="list-style-type: none"> <li>• Measured RSCP in dBm</li> </ul>
<i>ecio[Optional]</i>	<ul style="list-style-type: none"> <li>• Measured ECIO in dBm.</li> </ul>
<i>sinr[Optional]</i>	<ul style="list-style-type: none"> <li>• Measured SINR in dB. -15 dB is sent to clients if the actual SINR is less than -15 dB</li> </ul>

### 8.669.2 Field Documentation

8.669.2.1 float tdscdmaSigInfoExt::ecio

8.669.2.2 float tdscdmaSigInfoExt::rscp

8.669.2.3 float tdscdmaSigInfoExt::rssi

8.669.2.4 float tdscdmaSigInfoExt::sinr

## 8.670 tempData\_t Struct Reference

### Data Fields

- uint32\_t [timeSource](#)
- uint32\_t [timeOfFirstSample](#)
- uint8\_t [temperatureDataLen](#)
- uint16\_t [timeOffset](#) [64]
- uint32\_t [temperature](#) [64]

### 8.670.1 Detailed Description

This structure specifies information regarding the Temperature Data. Please check has\_<Param\_Name> field for presence of optional parameters

## 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>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>temperature-DataLen</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>

## 8.670.2 Field Documentation

8.670.2.1 uint32\_t tempData\_t::temperature[64]

8.670.2.2 uint8\_t tempData\_t::temperatureDataLen

8.670.2.3 uint32\_t tempData\_t::timeOfFirstSample

8.670.2.4 uint16\_t tempData\_t::timeOffset[64]

8.670.2.5 uint32\_t tempData\_t::timeSource

## 8.671 tmd\_mitigationDevList Struct Reference

## Data Fields

- uint8\_t [mitigationDevIdLen](#)
- char [mitigationDevId](#) [255]
- uint8\_t [maxMitigationLevel](#)

## 8.671.1 Detailed Description

This structure contains mitigation devices list



## 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>
<i>maxMitigationLevel</i>	<ul style="list-style-type: none"> <li>Maximum valid mitigation level.</li> <li>Valid range - 0 to max_mitigation_level.</li> </ul>

## 8.671.2 Field Documentation

8.671.2.1 uint8\_t tmd\_mitigationDevList::maxMitigationLevel

8.671.2.2 char tmd\_mitigationDevList::mitigationDevId[255]

8.671.2.3 uint8\_t tmd\_mitigationDevList::mitigationDevIdLen

## 8.672 transferRouteMessageTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- [sMSTransferRouteMTMessageInfo](#) [TransferRouteMTMessageInfo](#)

## 8.672.1 Detailed Description

This structure contains Transfer route MT message TLV information.

## 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>

## 8.672.2 Field Documentation

8.672.2.1 uint8\_t transferRouteMessageTlv::TlvPresent

8.672.2.2 [sMSTransferRouteMTMessageInfo](#) transferRouteMessageTlv::TransferRouteMTMessageInfo

## 8.673 uim\_additionalReadResult Struct Reference

## Data Fields

- uint16\_t [additionalRecordLen](#)
- uint8\_t [additionalRecord](#) [255]
- uint8\_t [TlvPresent](#)

### 8.673.1 Detailed Description

This structure contains the information for additional result.

#### Parameters

<i>additional-RecordLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of additionalRecord.</li> </ul>
<i>additional-Record[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Read content of all the additional records. Each record has the same size as the first record.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.673.2 Field Documentation

8.673.2.1 uint8\_t uim\_additionalReadResult::additionalRecord[255]

8.673.2.2 uint16\_t uim\_additionalReadResult::additionalRecordLen

8.673.2.3 uint8\_t uim\_additionalReadResult::TlvPresent

## 8.674 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](#)

### 8.674.1 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>

## 8.674.2 Field Documentation

8.674.2.1 uint8\_t uim\_appStatus::aidLength

8.674.2.2 uint8\_t uim\_appStatus::aidVal[255]

- 8.674.2.3 uint8\_t uim\_appStatus::appState
- 8.674.2.4 uint8\_t uim\_appStatus::appType
- 8.674.2.5 uint8\_t uim\_appStatus::persoFeature
- 8.674.2.6 uint8\_t uim\_appStatus::persoRetries
- 8.674.2.7 uint8\_t uim\_appStatus::persoState
- 8.674.2.8 uint8\_t uim\_appStatus::persoUnblockRetries
- 8.674.2.9 uint8\_t uim\_appStatus::pin1Retries
- 8.674.2.10 uint8\_t uim\_appStatus::pin1State
- 8.674.2.11 uint8\_t uim\_appStatus::pin2Retries
- 8.674.2.12 uint8\_t uim\_appStatus::pin2State
- 8.674.2.13 uint8\_t uim\_appStatus::puk1Retries
- 8.674.2.14 uint8\_t uim\_appStatus::puk2Retries
- 8.674.2.15 uint8\_t uim\_appStatus::univPin

## 8.675 uim\_authenticateResult Struct Reference

### Data Fields

- uint16\_t [contentLen](#)
- uint8\_t [content](#) [1024]

### 8.675.1 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[UIM_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>

### 8.675.2 Field Documentation

- 8.675.2.1 uint8\_t uim\_authenticateResult::content[1024]
- 8.675.2.2 uint16\_t uim\_authenticateResult::contentLen

## 8.676 uim\_authenticationData Struct Reference

### Data Fields

- uint8\_t [context](#)
- uint16\_t [dataLen](#)
- uint8\_t [data](#) [1024]

### 8.676.1 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[UIM_MAX- _DESCRIPTIO- N_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Authenticate Data.</li> </ul>

### 8.676.2 Field Documentation

8.676.2.1 uint8\_t uim\_authenticationData::context

8.676.2.2 uint8\_t uim\_authenticationData::data[1024]

8.676.2.3 uint16\_t uim\_authenticationData::dataLen

## 8.677 uim\_cardResult Struct Reference

### Data Fields

- uint8\_t [sw1](#)
- uint8\_t [sw2](#)

#### 8.677.1 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>

#### 8.677.2 Field Documentation

8.677.2.1 uint8\_t uim\_cardResult::sw1

8.677.2.2 uint8\_t uim\_cardResult::sw2

## 8.678 uim\_cardResultInfo Struct Reference

### Data Fields

- uint8\_t [sw1](#)
- uint8\_t [sw2](#)
- uint8\_t [TlvPresent](#)

#### 8.678.1 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>
<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• Tlv Present.</li></ul>

#### 8.678.2 Field Documentation

8.678.2.1 `uint8_t uim_cardResultInfo::sw1`

8.678.2.2 `uint8_t uim_cardResultInfo::sw2`

8.678.2.3 `uint8_t uim_cardResultInfo::TlvPresent`

## 8.679 `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]

### 8.679.1 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. <code>cardState</code></li> <li>• If zero(0) then no <code>cardState</code> information exists.</li> </ul>
<i>SlotInfo</i>	<ul style="list-style-type: none"> <li>• See <code>uim_slotInfo</code> for more information.</li> </ul>



## 8.679.2 Field Documentation

8.679.2.1 uint16\_t uim\_cardStatus::index1xPri

8.679.2.2 uint16\_t uim\_cardStatus::index1xSec

8.679.2.3 uint16\_t uim\_cardStatus::indexGwPri

8.679.2.4 uint16\_t uim\_cardStatus::indexGwSec

8.679.2.5 uint8\_t uim\_cardStatus::numSlot

8.679.2.6 uim\_slotInfo uim\_cardStatus::SlotInfo[5]

## 8.680 uim\_changeUIMPIN Struct Reference

### Data Fields

- uint8\_t [pinID](#)
- uint8\_t [oldPINLen](#)
- uint8\_t [oldPINVal](#) [255]
- uint8\_t [pinLen](#)
- uint8\_t [pinVal](#) [255]

### 8.680.1 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>

## 8.680.2 Field Documentation

8.680.2.1 uint8\_t uim\_changeUIMPIN::oldPINLen

8.680.2.2 uint8\_t uim\_changeUIMPIN::oldPINVal[255]

8.680.2.3 uint8\_t uim\_changeUIMPIN::pinID

8.680.2.4 uint8\_t uim\_changeUIMPIN::pinLen

8.680.2.5 uint8\_t uim\_changeUIMPIN::pinVal[255]

## 8.681 uim\_depersonalizationInformation Struct Reference

### Data Fields

- uint8\_t [feature](#)
- uint8\_t [operation](#)
- uint8\_t [ckLen](#)
- uint8\_t [ckVal](#) [255]

### 8.681.1 Detailed Description

This structure contains the Depersonalization Information.

#### Parameters

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

### 8.681.2 Field Documentation

8.681.2.1 `uint8_t uim_depersonalizationInformation::ckLen`

8.681.2.2 `uint8_t uim_depersonalizationInformation::ckVal[255]`

8.681.2.3 `uint8_t uim_depersonalizationInformation::feature`

8.681.2.4 `uint8_t uim_depersonalizationInformation::operation`

## 8.682 uim\_encryptedPIN1 Struct Reference

### Data Fields

- `uint8_t pin1Len`
- `uint8_t pin1Val [255]`

### 8.682.1 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.

### 8.682.2 Field Documentation

8.682.2.1 `uint8_t uim_encryptedPIN1::pin1Len`

8.682.2.2 `uint8_t uim_encryptedPIN1::pin1Val[255]`

## 8.683 uim\_fileAttributes Struct Reference

### Data Fields

- `uint16_t fileSize`
- `uint16_t fileID`
- `uint8_t fileType`
- `uint16_t recordSize`
- `uint16_t recordCount`
- `uint8_t secRead`
- `uint16_t secReadMask`
- `uint8_t secWrite`

- uint16\_t [secWriteMask](#)
- uint8\_t [secIncrease](#)
- uint16\_t [secIncreaseMask](#)
- uint8\_t [secDeactivate](#)
- uint16\_t [secDeactivateMask](#)
- uint8\_t [secActivate](#)
- uint16\_t [secActivateMask](#)
- uint16\_t [rawLen](#)
- uint8\_t [rawValue](#) [255]

### 8.683.1 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[MAX_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Raw value of file attributes.</li> </ul>

## 8.683.2 Field Documentation

8.683.2.1 uint16\_t uim\_fileAttributes::fileID

8.683.2.2 uint16\_t uim\_fileAttributes::fileSize

8.683.2.3 uint8\_t uim\_fileAttributes::fileType

8.683.2.4 uint16\_t uim\_fileAttributes::rawLen

8.683.2.5 uint8\_t uim\_fileAttributes::rawValue[255]

8.683.2.6 uint16\_t uim\_fileAttributes::recordCount

8.683.2.7 uint16\_t uim\_fileAttributes::recordSize

8.683.2.8 uint8\_t uim\_fileAttributes::secActivate

8.683.2.9 uint16\_t uim\_fileAttributes::secActivateMask

8.683.2.10 uint8\_t uim\_fileAttributes::secDeactivate

8.683.2.11 uint16\_t uim\_fileAttributes::secDeactivateMask

8.683.2.12 uint8\_t uim\_fileAttributes::secIncrease

8.683.2.13 uint16\_t uim\_fileAttributes::secIncreaseMask

8.683.2.14 uint8\_t uim\_fileAttributes::secRead

8.683.2.15 uint16\_t uim\_fileAttributes::secReadMask

8.683.2.16 uint8\_t uim\_fileAttributes::secWrite

8.683.2.17 uint16\_t uim\_fileAttributes::secWriteMask

## 8.684 uim\_fileInfo Struct Reference

### Data Fields

- uint16\_t [fileID](#)
- uint8\_t [pathLen](#)
- uint16\_t [path](#) [255]

### 8.684.1 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) is for LTE (0x3F00 ,0x7F20) is for GSM.</li> </ul>

### 8.684.2 Field Documentation

8.684.2.1 uint16\_t uim\_fileInfo::fileID

8.684.2.2 uint16\_t uim\_fileInfo::path[255]

8.684.2.3 uint8\_t uim\_fileInfo::pathLen

## 8.685 uim\_GetSlotsInfoTlv Struct Reference

### Data Fields

- uint8\_t [NumberOfPhySlotInfo](#)
- [uim\\_physlotInfo](#) [uimSlotInfo](#) [255]
- uint8\_t [TlvPresent](#)

### 8.685.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Information TLV.

## Parameters

<i>NumberOfPhy-SlotInfo</i>	<ul style="list-style-type: none"> <li>• Number of sets of the Slot Information.</li> </ul>
<i>uimSlotInfo</i>	<ul style="list-style-type: none"> <li>• Slots information See UIMSlotInfo for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Physical slot information TLV present.</li> </ul>

## 8.685.2 Field Documentation

8.685.2.1 uint8\_t uim\_GetSlotsInfoTlv::NumberOfPhySlotInfo

8.685.2.2 uint8\_t uim\_GetSlotsInfoTlv::TlvPresent

8.685.2.3 uim\_physlotInfo uim\_GetSlotsInfoTlv::uimSlotInfo[255]

## 8.686 uim\_GetSlotsStatusTlv Struct Reference

## Data Fields

- uint8\_t [NumberOfPhySlot](#)
- [slot\\_t uimSlotStatus](#) [255]
- uint8\_t [TlvPresent](#)

## 8.686.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status TLV.

## Parameters

<i>NumberOfPhy-Slot</i>	<ul style="list-style-type: none"> <li>• Number of sets of the Physical Slot Status.</li> </ul>
<i>uimSlotStatus</i>	<ul style="list-style-type: none"> <li>• Slots Status See UIMSlotStatus for more information.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Physical slot status TLV present.</li> </ul>

## 8.686.2 Field Documentation

8.686.2.1 uint8\_t uim\_GetSlotsStatusTlv::NumberOfPhySlot

8.686.2.2 uint8\_t uim\_GetSlotsStatusTlv::TlvPresent

8.686.2.3 slot\_t uim\_GetSlotsStatusTlv::uimSlotStatus[255]



## 8.687 uim\_hotSwapStatus Struct Reference

### Data Fields

- uint8\_t [hotSwapLength](#)
- uint8\_t [hotSwap](#) [255]

### 8.687.1 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>

### 8.687.2 Field Documentation

8.687.2.1 uint8\_t uim\_hotSwapStatus::hotSwap[255]

8.687.2.2 uint8\_t uim\_hotSwapStatus::hotSwapLength

## 8.688 uim\_indToken Struct Reference

### Data Fields

- uint32\_t [token](#)
- uint8\_t [TlvPresent](#)

### 8.688.1 Detailed Description

This structure contains the information for indication.

#### Parameters

<i>pIndicationToken</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>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

### 8.688.2 Field Documentation

8.688.2.1 uint8\_t uim\_indToken::TlvPresent

8.688.2.2 uint32\_t uim\_indToken::token

## 8.689 uim\_personalizationStatus Struct Reference

### Data Fields

- uint8\_t [numFeatures](#)
- uint8\_t [feature](#) [12]
- uint8\_t [verifyLeft](#) [12]
- uint8\_t [unblockLeft](#) [12]

### 8.689.1 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>

### 8.689.2 Field Documentation

8.689.2.1 uint8\_t uim\_personalizationStatus::feature[12]

8.689.2.2 uint8\_t uim\_personalizationStatus::numFeatures

8.689.2.3 uint8\_t uim\_personalizationStatus::unblockLeft[12]

8.689.2.4 uint8\_t uim\_personalizationStatus::verifyLeft[12]

## 8.690 uim\_physlotInfo Struct Reference

### Data Fields

- uint32\_t [cardProtocol](#)
- uint8\_t [numApp](#)
- uint8\_t [atrValueLen](#)
- uint8\_t [atrValue](#) [255]
- uint8\_t [iseUICC](#)

### 8.690.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>cardProtocol</i>	<ul style="list-style-type: none"> <li>• Protocol of the card.</li> <li>• Valid values: <ul style="list-style-type: none"> <li>– 0x00 - Unknown</li> <li>– 0x01 - ICC Protocol</li> <li>– 0x02 - UICC Protocol</li> </ul> </li> </ul>
<i>numApp</i>	<ul style="list-style-type: none"> <li>• Number of valid applications present in the EF-DIR of card.</li> </ul>
<i>atrValueLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the atrValue</li> </ul>
<i>atrValue</i>	<ul style="list-style-type: none"> <li>• Raw value of the ATR sent by the card during the initialization.</li> </ul>
<i>iseUICC</i>	<ul style="list-style-type: none"> <li>• Indicates whether the card is an eUICC card based on the ATR.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Not an eUICC card</li> <li>– 1 - eUICC card</li> </ul> </li> </ul>

### 8.690.2 Field Documentation

8.690.2.1 uint8\_t uim\_physlotInfo::atrValue[255]

8.690.2.2 uint8\_t uim\_physlotInfo::atrValueLen

8.690.2.3 uint32\_t uim\_physlotInfo::cardProtocol

8.690.2.4 uint8\_t uim\_physlotInfo::iseUICC

8.690.2.5 uint8\_t uim\_physlotInfo::numApp

## 8.691 uim\_physlotsInfo Struct Reference

## Data Fields

- [uim\\_physlotInfo](#) [uimSlotInfo](#) [255]

### 8.691.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>uimSlotInfo</i> [MAX_PHY_SLOTS_INFO]	<ul style="list-style-type: none"> <li>• Contain all slots Information.</li> </ul>
---	--

### 8.691.2 Field Documentation

8.691.2.1 [uim\\_physlotInfo](#) [uim\\_physlotsInfo::uimSlotInfo](#)[255]

## 8.692 [uim\\_readRecordInfo](#) Struct Reference

## Data Fields

- [uint16\\_t](#) [record](#)
- [uint16\\_t](#) [length](#)

### 8.692.1 Detailed Description

This structure contains the information for read record information.

#### Parameters

<i>record</i>	<ul style="list-style-type: none"> <li>• Record number (starting from 1).</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 record.</li> </ul>

### 8.692.2 Field Documentation

8.692.2.1 [uint16\\_t](#) [uim\\_readRecordInfo::length](#)

8.692.2.2 [uint16\\_t](#) [uim\\_readRecordInfo::record](#)

## 8.693 [uim\\_readResult](#) Struct Reference

## Data Fields

- [uint16\\_t](#) [contentLen](#)
- [uint8\\_t](#) [content](#) [255]

### 8.693.1 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[255]</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>

### 8.693.2 Field Documentation

8.693.2.1 `uint8_t uim_readResult::content[255]`

8.693.2.2 `uint16_t uim_readResult::contentLen`

## 8.694 uim\_readResultInfo Struct Reference

### Data Fields

- `uint16_t` [contentLen](#)
- `uint8_t` [content](#) [255]
- `uint8_t` [TlvPresent](#)

### 8.694.1 Detailed Description

This structure contains the information for read operation.

#### Parameters

<i>contentLen</i>	<ul style="list-style-type: none"> <li>Number of sets of content.</li> </ul>
<i>content[ 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>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>Tlv Present.</li> </ul>

### 8.694.2 Field Documentation

8.694.2.1 `uint8_t uim_readResultInfo::content[255]`

8.694.2.2 `uint16_t uim_readResultInfo::contentLen`

8.694.2.3 `uint8_t uim_readResultInfo::TlvPresent`

## 8.695 uim\_readTransparentInfo Struct Reference

### Data Fields

- uint16\_t [offset](#)
- uint16\_t [length](#)

### 8.695.1 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>

### 8.695.2 Field Documentation

8.695.2.1 uint16\_t uim\_readTransparentInfo::length

8.695.2.2 uint16\_t uim\_readTransparentInfo::offset

## 8.696 uim\_refreshevent Struct Reference

### Data Fields

- uint8\_t [stage](#)
- uint8\_t [mode](#)
- uint8\_t [sessionType](#)
- uint8\_t [aidLength](#)
- uint8\_t [aid](#) [255]
- uint16\_t [numOfFiles](#)
- [uim\\_fileInfo arrfileInfo](#) [255]

### 8.696.1 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>

## 8.696.2 Field Documentation

8.696.2.1 `uint8_t uim_refreshevent::aid[255]`

8.696.2.2 `uint8_t uim_refreshevent::aidLength`

8.696.2.3 `uim_fileInfo uim_refreshevent::arrfileInfo[255]`

8.696.2.4 `uint8_t uim_refreshevent::mode`

8.696.2.5 `uint16_t uim_refreshevent::numOfFiles`

8.696.2.6 `uint8_t uim_refreshevent::sessionType`

8.696.2.7 `uint8_t uim_refreshevent::stage`

## 8.697 `uim_registerRefresh` Struct Reference

### Data Fields

- `uint8_t registerFlag`
- `uint8_t voteForInit`
- `uint16_t numFiles`
- `uim_fileInfo arrfileInfo` [255]

### 8.697.1 Detailed Description

This structure contains parameters 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 <a href="#">uim_fileInfo</a> for more information</li> </ul>

### 8.697.2 Field Documentation

8.697.2.1 `uim_fileInfo uim_registerRefresh::arrfileInfo`[255]

8.697.2.2 `uint16_t uim_registerRefresh::numFiles`

8.697.2.3 `uint8_t uim_registerRefresh::registerFlag`

8.697.2.4 `uint8_t uim_registerRefresh::voteForInit`



## 8.698 uim\_remainingRetries Struct Reference

### Data Fields

- uint8\_t [verifyLeft](#)
- uint8\_t [unblockLeft](#)

### 8.698.1 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.

### 8.698.2 Field Documentation

8.698.2.1 uint8\_t uim\_remainingRetries::unblockLeft

8.698.2.2 uint8\_t uim\_remainingRetries::verifyLeft

## 8.699 uim\_sessionInformation Struct Reference

### Data Fields

- uint8\_t [sessionType](#)
- uint8\_t [aidLength](#)
- uint8\_t [aid](#) [255]

### 8.699.1 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>

## 8.699.2 Field Documentation

8.699.2.1 uint8\_t uim\_sessionInformation::aid[255]

8.699.2.2 uint8\_t uim\_sessionInformation::aidLength

8.699.2.3 uint8\_t uim\_sessionInformation::sessionType

## 8.700 uim\_setPINProtection Struct Reference

## Data Fields

- uint8\_t [pinID](#)
- uint8\_t [pinOperation](#)
- uint8\_t [pinLength](#)
- uint8\_t [pinValue](#) [255]

## 8.700.1 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>

## 8.700.2 Field Documentation

8.700.2.1 uint8\_t uim\_setPINProtection::pinID

8.700.2.2 uint8\_t uim\_setPINProtection::pinLength

8.700.2.3 uint8\_t uim\_setPINProtection::pinOperation

8.700.2.4 uint8\_t uim\_setPINProtection::pinValue[255]

## 8.701 uim\_simBusyStatus Struct Reference

### Data Fields

- uint8\_t [simBusyLength](#)
- uint8\_t [simBusy](#) [255]

### 8.701.1 Detailed Description

This structure contains Sim Busy Status Information.

#### Parameters

<i>simBusyLength</i>	<ul style="list-style-type: none"> <li>Number of sets of the following elements. i.e. simBusy</li> </ul>
<i>simBusy</i> [MAX_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>Indicates the status of the valid card. <ul style="list-style-type: none"> <li>0 - Sim card is not busy</li> <li>1 - Sim card is busy</li> </ul> </li> </ul>

## 8.701.2 Field Documentation

8.701.2.1 uint8\_t uim\_simBusyStatus::simBusy[255]

8.701.2.2 uint8\_t uim\_simBusyStatus::simBusyLength

## 8.702 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]

### 8.702.1 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>

## 8.702.2 Field Documentation

8.702.2.1 `uim_appStatus uim_slotInfo::AppStatus[10]`

8.702.2.2 `uint8_t uim_slotInfo::cardState`

8.702.2.3 `uint8_t uim_slotInfo::errorState`

8.702.2.4 `uint8_t uim_slotInfo::numApp`

8.702.2.5 `uint8_t uim_slotInfo::upinRetries`

8.702.2.6 `uint8_t uim_slotInfo::upinState`

8.702.2.7 `uint8_t uim_slotInfo::upukRetries`

## 8.703 uim\_UIMGetFDNStatus Struct Reference

### Data Fields

- `uint8_t` [FDNStatus](#)
- `uint8_t` [TlvPresent](#)

### 8.703.1 Detailed Description

This structure contains the information for FDN Status.

## Parameters

<i>FDNStatus</i>	0 – FDN is not available 1 – FDN is available and disabled 2 – FDN is available and enabled
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.703.2 Field Documentation

8.703.2.1 uint8\_t uim\_UIMGetFDNStatus::FDNStatus

8.703.2.2 uint8\_t uim\_UIMGetFDNStatus::TlvPresent

## 8.704 uim\_UIMGetHiddenKeyStatus Struct Reference

## Data Fields

- uint8\_t [hiddenKey](#)
- uint8\_t [TlvPresent](#)

## 8.704.1 Detailed Description

This structure contains the information for Hidden Key Status.

## Parameters

<i>hiddenKey</i>	<ul style="list-style-type: none"> <li>• Status of hidden key PIN 0 – Not supported 1 – Enabled and not verified 2 – Enabled and verified 3 – Disabled</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.704.2 Field Documentation

8.704.2.1 uint8\_t uim\_UIMGetHiddenKeyStatus::hiddenKey

8.704.2.2 uint8\_t uim\_UIMGetHiddenKeyStatus::TlvPresent

## 8.705 uim\_UIMGetIndex Struct Reference

## Data Fields

- uint8\_t [index](#)
- uint8\_t [TlvPresent](#)

## 8.705.1 Detailed Description

This structure contains the information for getting Index.

## Parameters

<i>index</i>	<ul style="list-style-type: none"> <li>• Index of the application in EF_DIR file, starting from 1.</li> </ul>
<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• Tlv Present.</li> </ul>

## 8.705.2 Field Documentation

8.705.2.1 uint8\_t uim\_UIMGetIndex::index

8.705.2.2 uint8\_t uim\_UIMGetIndex::TlvPresent

## 8.706 uim\_UIMSessionInformation Struct Reference

## Data Fields

- uint8\_t [sessionType](#)
- uint8\_t [aidLength](#)
- uint8\_t [aid](#) [255]

## 8.706.1 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>

## 8.706.2 Field Documentation

8.706.2.1 `uint8_t uim_UIMSessionInformation::aid[255]`

8.706.2.2 `uint8_t uim_UIMSessionInformation::aidLength`

8.706.2.3 `uint8_t uim_UIMSessionInformation::sessionType`

## 8.707 uim\_unblockUIMPIN Struct Reference

### Data Fields

- `uint8_t pinID`
- `uint8_t pukLen`
- `uint8_t pukVal [255]`
- `uint8_t newPINLen`
- `uint8_t newPINVal [255]`

### 8.707.1 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>

### 8.707.2 Field Documentation

8.707.2.1 `uint8_t uim_unblockUIMPIN::newPINLen`

8.707.2.2 `uint8_t uim_unblockUIMPIN::newPINVal[255]`

8.707.2.3 `uint8_t uim_unblockUIMPIN::pinID`

8.707.2.4 `uint8_t uim_unblockUIMPIN::pukLen`



8.707.2.5 uint8\_t uim\_unblockUIMPIN::pukVal[255]

## 8.708 uim\_validCardStatus Struct Reference

### Data Fields

- uint8\_t [validCardLength](#)
- uint8\_t [validCard](#) [255]

### 8.708.1 Detailed Description

This structure contains Valid Card Status Information.

#### Parameters

<i>validCardLength</i>	<ul style="list-style-type: none"> <li>• Number of sets of the following elements. i.e. validCard</li> </ul>
<i>validCard[<a href="#">MAX_DESCRIPTION_LENGTH</a>]</i>	<ul style="list-style-type: none"> <li>• Indicates the status of the valid card. <ul style="list-style-type: none"> <li>– 0 - Status of the card is unknown</li> <li>– 1 - Status of the card is valid</li> </ul> </li> </ul>

### 8.708.2 Field Documentation

8.708.2.1 uint8\_t uim\_validCardStatus::validCard[255]

8.708.2.2 uint8\_t uim\_validCardStatus::validCardLength

## 8.709 uim\_verifyUIMPIN Struct Reference

### Data Fields

- uint8\_t [pinID](#)
- uint8\_t [pinLen](#)
- uint8\_t [pinVal](#) [255]

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

## 8.709.2 Field Documentation

8.709.2.1 `uint8_t uim_verifyUIMPIN::pinID`

8.709.2.2 `uint8_t uim_verifyUIMPIN::pinLen`

8.709.2.3 `uint8_t uim_verifyUIMPIN::pinVal[255]`

## 8.710 `uim_writeRecordInfo` Struct Reference

### Data Fields

- `uint16_t` [record](#)
- `uint16_t` [dataLen](#)
- `uint8_t` [data](#) [255]

### 8.710.1 Detailed Description

This structure contains the information for write operation.

#### Parameters

<i>record</i>	<ul style="list-style-type: none"> <li>Record number (starting from 1).</li> <li>This field is ignored for cyclic files.</li> </ul>
<i>length</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>Content to write.</li> </ul>

## 8.710.2 Field Documentation

8.710.2.1 `uint8_t uim_writeRecordInfo::data[255]`

8.710.2.2 `uint16_t uim_writeRecordInfo::dataLen`

8.710.2.3 `uint16_t uim_writeRecordInfo::record`

## 8.711 `uim_writeTransparentInfo` Struct Reference

## Data Fields

- uint16\_t [offset](#)
- uint16\_t [dataLen](#)
- uint8\_t [data](#) [1024]

### 8.711.1 Detailed Description

This structure contains the information for write operation.

#### Parameters

<i>offset</i>	<ul style="list-style-type: none"> <li>• Offset for the write operation.</li> </ul>
<i>dataLen</i>	<ul style="list-style-type: none"> <li>• Length of the following elements i.e. data.</li> </ul>
<i>data[<a href="#">MAX_CON-</a> <a href="#">TENT_LENGTH</a>]</i>	<ul style="list-style-type: none"> <li>• Content to write.</li> </ul>

### 8.711.2 Field Documentation

8.711.2.1 [uint8\\_t uim\\_writeTransparentInfo::data\[1024\]](#)

8.711.2.2 [uint16\\_t uim\\_writeTransparentInfo::dataLen](#)

8.711.2.3 [uint16\\_t uim\\_writeTransparentInfo::offset](#)

## 8.712 unpack\_audio\_SLQSGetAudioPathConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pECMode](#)
- uint8\_t \* [pNSEnable](#)
- uint16\_t \* [pTXGain](#)
- uint16\_t \* [pDTMFTXGain](#)
- uint16\_t \* [pCodecSTGain](#)
- [audio\\_TXPCMIIRFiltr](#) \* [pTXPCMIIRFiltr](#)
- [audio\\_RXPCMIIRFiltr](#) \* [pRXPCMIIRFiltr](#)
- uint8\_t \* [pMICGainSelect](#)
- uint8\_t \* [pRXAVCAGCSwitch](#)
- uint8\_t \* [pTXAVCSwitch](#)
- [audio\\_RXAGCList](#) \* [pRXAGCList](#)
- [audio\\_RXAVCList](#) \* [pRXAVCList](#)
- [audio\\_TXAGCList](#) \* [pTXAGCList](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.712.1 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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></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> <li>• Bit to check in ParamPresenceMask - <b>36</b></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> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>pTXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">audio_TXPCMIIRFiltr</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pRXPCMIIRFiltr</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">audio_RXPCMIIRFiltr</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>pMICGainSelect</i>	[Optional] <ul style="list-style-type: none"> <li>• AV_MICGAIN</li> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>pRXAVCAGC-Switch</i>	[Optional] <ul style="list-style-type: none"> <li>• RX AVC/AGC Switch</li> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>pTXAVCSwitch</i>	[Optional] <ul style="list-style-type: none"> <li>• TX AVC Switch</li> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>

<i>pRXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">audio_RXAGCList</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>43</b></li> </ul>
<i>pRXAVCList</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">audio_RXAVCList</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>44</b></li> </ul>
<i>pTXAGCList</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">audio_TXAGCList</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>45</b></li> </ul>

## 8.712.2 Field Documentation

8.712.2.1 `swi_uint256_t unpack_audio_SLQSGetAudioPathConfig_t::ParamPresenceMask`

8.712.2.2 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pCodecSTGain`

8.712.2.3 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pDTMFTXGain`

8.712.2.4 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pECMode`

8.712.2.5 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pMICGainSelect`

8.712.2.6 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pNSEnable`

8.712.2.7 `audio_RXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAGCList`

8.712.2.8 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCAGCSwitch`

8.712.2.9 `audio_RXAVCList* unpack_audio_SLQSGetAudioPathConfig_t::pRXAVCList`

8.712.2.10 `audio_RXPCMIIRFiltr* unpack_audio_SLQSGetAudioPathConfig_t::pRXPCMIIRFiltr`

8.712.2.11 `audio_TXAGCList* unpack_audio_SLQSGetAudioPathConfig_t::pTXAGCList`

8.712.2.12 `uint8_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXAVCSwitch`

8.712.2.13 `uint16_t* unpack_audio_SLQSGetAudioPathConfig_t::pTXGain`

8.712.2.14 `audio_TXPCMIIRFiltr* unpack_audio_SLQSGetAudioPathConfig_t::pTXPCMIIRFiltr`

## 8.713 unpack\_audio\_SLQSGetAudioProfile\_t Struct Reference

### Data Fields

- `uint8_t Profile`
- `uint8_t EarMute`
- `uint8_t MicMute`
- `uint8_t Volume`
- `swi_uint256_t ParamPresenceMask`

### 8.713.1 Detailed Description

This structure contains the unpack parameters to Get Audio Profile.

#### 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>

### 8.713.2 Field Documentation

8.713.2.1 uint8\_t unpack\_audio\_SLQSGetAudioProfile\_t::EarMute

8.713.2.2 uint8\_t unpack\_audio\_SLQSGetAudioProfile\_t::MicMute

8.713.2.3 swi\_uint256\_t unpack\_audio\_SLQSGetAudioProfile\_t::ParamPresenceMask

8.713.2.4 uint8\_t unpack\_audio\_SLQSGetAudioProfile\_t::Profile

8.713.2.5 uint8\_t unpack\_audio\_SLQSGetAudioProfile\_t::Volume

## 8.714 unpack\_audio\_SLQSGetAudioVolTLBConfig\_t Struct Reference

## Data Fields

- uint16\_t [ResCode](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.714.1 Detailed Description

This structure contains the unpack parameters for SLQSGetAudioVoITLBConfig.

## Parameters

<i>ResCode</i>	<ul style="list-style-type: none"> <li>• Result of requested item</li> <li>• Bit to check in ParamPresenceMask - <b>33</b></li> </ul>
----------------	---

## 8.714.2 Field Documentation

8.714.2.1 swi\_uint256\_t unpack\_audio\_SLQSGetAudioVoITLBConfig\_t::ParamPresenceMask

8.714.2.2 uint16\_t unpack\_audio\_SLQSGetAudioVoITLBConfig\_t::ResCode

## 8.715 unpack\_audio\_SLQSSetAudioVoITLBConfig\_t Struct Reference

## Data Fields

- uint16\_t [ResCode](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.715.1 Detailed Description

This structure contains the unpack parameters for SLQSSetAudioVoITLBConfig.

## Parameters

<i>ResCode</i>	<ul style="list-style-type: none"> <li>• Result of requested item</li> <li>• Bit to check in ParamPresenceMask - <b>33</b></li> </ul>
----------------	---

## 8.715.2 Field Documentation

8.715.2.1 swi\_uint256\_t unpack\_audio\_SLQSSetAudioVoITLBConfig\_t::ParamPresenceMask

8.715.2.2 uint16\_t unpack\_audio\_SLQSSetAudioVoITLBConfig\_t::ResCode

## 8.716 unpack\_cat\_SetCatEventCallback\_ind\_t Struct Reference

## Data Fields

- uint8\_t [event\\_Index](#)
- struct [cat\\_commonEventTlv](#) [CCETlv](#) [11]

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.716.1 Detailed Description

Structure used to store all CAT Notification Parameters.

#### Parameters

<i>event_Index</i>	- Event Index
<i>CCETlv</i>	- CAT common event Tlv <a href="#">cat_commonEventTlv</a> for more info. <ul style="list-style-type: none"> <li>• [0] - Bit to check in ParamPresenceMask for Display Text - <b>16</b></li> <li>• [1] - Bit to check in ParamPresenceMask for Get In-Key - <b>17</b></li> <li>• [2] - Bit to check in ParamPresenceMask for Get Input - <b>18</b></li> <li>• [3] - Bit to check in ParamPresenceMask for Setup Menu - <b>19</b></li> <li>• [4] - Bit to check in ParamPresenceMask for Select Item - <b>20</b></li> <li>• [5] - Bit to check in ParamPresenceMask for Alpha Identifier - <b>21</b></li> <li>• [6] - Bit to check in ParamPresenceMask for Setup Event List - <b>22</b></li> <li>• [7] - Bit to check in ParamPresenceMask for Setup Idle Mode Text - <b>23</b></li> <li>• [8] - Bit to check in ParamPresenceMask for Language Notification - <b>24</b></li> <li>• [9] - Bit to check in ParamPresenceMask for Refresh - <b>25</b></li> <li>• [10] - Bit to check in ParamPresenceMask for End Proactive Session - <b>26</b></li> </ul>

### 8.716.2 Field Documentation

8.716.2.1 `struct cat_commonEventTlv unpack_cat_SetCatEventCallback_ind_t::CCETlv[11]`

8.716.2.2 `uint8_t unpack_cat_SetCatEventCallback_ind_t::event_Index`

8.716.2.3 `swi_uint256_t unpack_cat_SetCatEventCallback_ind_t::ParamPresenceMask`

## 8.717 unpack\_cat\_SetCATEventCallback\_t Struct Reference

### Data Fields

- `uint32_t` [errorMask](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.717.1 Detailed Description

This structure contains sEnables/disables the CAT event callback unpack variable.



## Parameters

<i>errorMask</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

## 8.717.2 Field Documentation

8.717.2.1 uint32\_t unpack\_cat\_SetCATEventCallback\_t::errorMask

8.717.2.2 swi\_uint256\_t unpack\_cat\_SetCATEventCallback\_t::ParamPresenceMask

8.717.2.3 uint16\_t unpack\_cat\_SetCATEventCallback\_t::Tlvresult

## 8.718 unpack\_dms\_GetActivationState\_t Struct Reference

## Data Fields

- [uint8\\_t state](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

## 8.718.1 Detailed Description

This structure contains unpack\_dms\_GetActivationState parameters.

## Parameters

<i>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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.718.2 Field Documentation

8.718.2.1 `swi_uint256_t` `unpack_dms_GetActivationState_t::ParamPresenceMask`8.718.2.2 `uint8_t` `unpack_dms_GetActivationState_t::state`8.719 `unpack_dms_GetBandCapability_t` Struct Reference

## Data Fields

- `uint64_t` [BandCapability](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.719.1 Detailed Description

Structure to store the band capability of the device.

## Parameters

<i>BandCapability</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> <li>• Bit 60 - WCDMA Japan 850 band</li> <li>• Bit 61 - WCDMA 1500 band</li> <li>• Bits 62 and 63 - Reserved</li> </ul>
-----------------------	---

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.719.2 Field Documentation

8.719.2.1 uint64\_t unpack\_dms\_GetBandCapability\_t::BandCapability

8.719.2.2 swi\_uint256\_t unpack\_dms\_GetBandCapability\_t::ParamPresenceMask

8.719.2.3 uint16\_t unpack\_dms\_GetBandCapability\_t::Tlvresult

## 8.720 unpack\_dms\_GetCrashAction\_t Struct Reference

## Data Fields

- uint8\_t [DevCrashState](#)
- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.720.1 Detailed Description

This structure is used to store the Crash State from the device.

## Parameters

<i>DevCrashState</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 SLQSSwiGetCrashInfo() SDK API</li> <li>2 - No action</li> </ul> </li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.720.2 Field Documentation

8.720.2.1 uint8\_t unpack\_dms\_GetCrashAction\_t::DevCrashState

8.720.2.2 swi\_uint256\_t unpack\_dms\_GetCrashAction\_t::ParamPresenceMask

8.720.2.3 uint16\_t unpack\_dms\_GetCrashAction\_t::Tlvresult

## 8.721 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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.721.1 Detailed Description

This structure contains current settings of custom features

#### Parameters

<i>GpsEnable</i>	<ul style="list-style-type: none"> <li>• describes if GPS is enabled or disabled</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - GPS is disabled</li> <li>– 0x01 - GPS is enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>DisableIMSI</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>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>IPFamSupport</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>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>RMAutoConnect</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>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>GPSSel</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 Port</li> <li>– 0x01 - GPS Rx over AUX Port</li> <li>– 0x02 - GPS Rx over dedicated GPS port with no bias voltage applied</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>SMSSupport</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>• Bit to check in ParamPresenceMask - <b>21</b></li> <li>• Used to determine whether or not to hide SMS from user</li> </ul>
<i>IsVoiceEnabled</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> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>DHCPRelay-Enabled</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• DHCP Relay support</li> <li>• values: <ul style="list-style-type: none"> <li>– 0x00 - Disable DHCP relay</li> <li>– 0x01 - Enable DHCP relay</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>GPSSLPM</i>	<ul style="list-style-type: none"> <li>• optional 1 byte parameter</li> <li>• GPSSLPM 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> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv Result.</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.721.2 Field Documentation

8.721.2.1 uint8\_t unpack\_dms\_GetCustFeature\_t::DHCPRelayEnabled

8.721.2.2 uint8\_t unpack\_dms\_GetCustFeature\_t::DisableIMSI

8.721.2.3 uint32\_t unpack\_dms\_GetCustFeature\_t::GpsEnable

8.721.2.4 uint8\_t unpack\_dms\_GetCustFeature\_t::GPSLPM

8.721.2.5 uint8\_t unpack\_dms\_GetCustFeature\_t::GPSSel

8.721.2.6 uint16\_t unpack\_dms\_GetCustFeature\_t::IPFamSupport

8.721.2.7 uint8\_t unpack\_dms\_GetCustFeature\_t::IsVoiceEnabled

8.721.2.8 swi\_uint256\_t unpack\_dms\_GetCustFeature\_t::ParamPresenceMask

8.721.2.9 uint8\_t unpack\_dms\_GetCustFeature\_t::RMAutoConnect

8.721.2.10 uint8\_t unpack\_dms\_GetCustFeature\_t::SMSSupport

8.721.2.11 uint16\_t unpack\_dms\_GetCustFeature\_t::Tlvresult

## 8.722 unpack\_dms\_GetCustFeaturesV2\_t Struct Reference

### Data Fields

- [DMSgetCustomFeatureV2 GetCustomFeatureV2](#)
- uint16\_t *Tlvresult*
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.722.1 Detailed Description

This structure contains customization settings set to modem unpack

#### Parameters

<i>GetCustomFeatureV2</i>	<ul style="list-style-type: none"> <li>See <a href="#">DMSgetCustomFeatureV2</a> for more information</li> <li>pCustSettingInfo <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>32</b></li> </ul> </li> <li>pCustSettingList <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>33</b></li> </ul> </li> </ul>
---------------------------	--

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.722.2 Field Documentation

8.722.2.1 DMSgetCustomFeatureV2 unpack\_dms\_GetCustFeaturesV2\_t::GetCustomFeatureV2

8.722.2.2 swi\_uint256\_t unpack\_dms\_GetCustFeaturesV2\_t::ParamPresenceMask

8.722.2.3 uint16\_t unpack\_dms\_GetCustFeaturesV2\_t::Tlvresult

## 8.723 unpack\_dms\_GetDeviceCap\_t Struct Reference

### Data Fields

- uint32\_t [MaxTXChannelRate](#)
- uint32\_t [MaxRXChannelRate](#)
- uint32\_t [DataServiceCapability](#)
- uint32\_t [SimCapability](#)
- uint32\_t [RadiofacesSize](#)
- uint8\_t [Radiofaces](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.723.1 Detailed Description

Structure to store the device capabilities

#### Parameters

<i>MaxTXChannelRate</i>	<ul style="list-style-type: none"> <li>• Maximum transmission rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>MaxRXChannelRate</i>	<ul style="list-style-type: none"> <li>• Maximum reception rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>DataServiceCapability</i>	<ul style="list-style-type: none"> <li>• CS/PS data service capability <ul style="list-style-type: none"> <li>– 0 - No data services supported</li> <li>– 1 - Only Circuit Switched (CS) services supported</li> <li>– 2 - Only Packet Switched (PS) services supported</li> <li>– 3 - Simultaneous CS and PS</li> <li>– 4 - Non-simultaneous CS and PS</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>



<i>SimCapability</i>	Device SIM capability <ul style="list-style-type: none"> <li>• 0 - SIM not supported</li> <li>• 1 - SIM supported</li> </ul>
----------------------	--

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>RadiofacesSize</i>	<ul style="list-style-type: none"> <li>• Upon input, the maximum number of elements that the radio interface array can contain</li> <li>• Upon successful output, actual number of elements in the radio interface array</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Radiofaces</i>	<ul style="list-style-type: none"> <li>• Radio interface array. This is a structure of array containing the elements below. uint8_t Radiofaces               <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableRadioInterfaces.h</a> for Radio Interfaces</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.723.2 Field Documentation

8.723.2.1 uint32\_t unpack\_dms\_GetDeviceCap\_t::DataServiceCapability

8.723.2.2 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxRXChannelRate

8.723.2.3 uint32\_t unpack\_dms\_GetDeviceCap\_t::MaxTXChannelRate

8.723.2.4 swi\_uint256\_t unpack\_dms\_GetDeviceCap\_t::ParamPresenceMask

8.723.2.5 uint8\_t unpack\_dms\_GetDeviceCap\_t::Radiofaces[255]

8.723.2.6 uint32\_t unpack\_dms\_GetDeviceCap\_t::RadiofacesSize

8.723.2.7 uint32\_t unpack\_dms\_GetDeviceCap\_t::SimCapability

8.723.2.8 uint16\_t unpack\_dms\_GetDeviceCap\_t::Tlvresult

## 8.724 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]
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.724.1 Detailed Description

Gets the device capabilities structure

#### Parameters

<i>maxTxChannel-Rate</i>	<ul style="list-style-type: none"> <li>• Maximum transmission rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>maxRxChannel-Rate</i>	<ul style="list-style-type: none"> <li>• Maximum reception rate (in bps) supported by the device</li> <li>• In multi-technology devices, this value will be the greatest rate among all supported technologies</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>dataServiceCa-Capability</i>	<ul style="list-style-type: none"> <li>• CS/PS data service capability <ul style="list-style-type: none"> <li>– 0 - No data services supported</li> <li>– 1 - Only Circuit Switched (CS) services supported</li> <li>– 2 - Only Packet Switched (PS) services supported</li> <li>– 3 - Simultaneous CS and PS</li> <li>– 4 - Non-simultaneous CS and PS</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>simCapability</i>	<ul style="list-style-type: none"> <li>• Device SIM capability <ul style="list-style-type: none"> <li>– 0 - SIM not supported</li> <li>– 1 - SIM supported</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>radiolfacesSize</i>	<ul style="list-style-type: none"> <li>• Upon input, the maximum number of elements that the radio interface array can contain</li> <li>• Upon successful output, actual number of elements in the radio interface array</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Radiolfaces[OUT]</i>	<ul style="list-style-type: none"> <li>• Radio interface array. This is a structure of array containing the elements below. <ul style="list-style-type: none"> <li>– See <a href="#">qaGobiApiTableRadioInterfaces.h</a> for Radio Interfaces</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.724.2 Field Documentation

8.724.2.1 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::dataServiceCaCapability

8.724.2.2 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::maxRxChannelRate

8.724.2.3 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::maxTxChannelRate

8.724.2.4 swi\_uint256\_t unpack\_dms\_GetDeviceCapabilities\_t::ParamPresenceMask

8.724.2.5 uint8\_t unpack\_dms\_GetDeviceCapabilities\_t::Radiofaces[255]

8.724.2.6 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::radiofacesSize

8.724.2.7 uint32\_t unpack\_dms\_GetDeviceCapabilities\_t::simCapability

## 8.725 unpack\_dms\_GetDeviceCapabilitiesV2\_t Struct Reference

### Data Fields

- [dms\\_devCaps](#) DevCaps
- uint32\_t \* [pDevSrvCaps](#)
- uint64\_t \* [pDevVoiceCaps](#)
- uint64\_t \* [pDevVoiceDataCaps](#)
- [dms\\_devMultiSimCaps](#) \* [pDevMultiSimCaps](#)
- [dms\\_devMultiSimVoiceDataCaps](#) \* [pDevMultiSimVoiceDataCaps](#)
- [dms\\_devCurSubsCaps](#) \* [pDevCurSubsCaps](#)
- [dms\\_devSubsVoiceDataCaps](#) \* [pDevSubsVoiceDataCaps](#)
- [dms\\_devSubsFeatureModeCaps](#) \* [pDevSubsFeatureModeCaps](#)
- uint8\_t \* [pDevMaxActDataSubsCaps](#)
- [dms\\_devMaxSubsCaps](#) \* [pDevMaxSubsCaps](#)
- [dms\\_devMaxCfgListCaps](#) \* [pDevMaxCfgListCaps](#)
- int16\_t \* [pDevExplicitCfgIndex](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.725.1 Detailed Description

This structure contains the unpack\_dms\_GetDeviceCapabilitiesV2 response parameters.

#### Parameters

<i>DevCaps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pDevSrvCaps</i>	<ul style="list-style-type: none"> <li>• Device service capabilities             <ul style="list-style-type: none"> <li>– DMS_DEVICE_CAP_DATA_ONLY -0x01</li> <li>– DMS_DEVICE_CAP_VOICE_ONLY -0x02</li> <li>– DMS_DEVICE_CAP_SIMUL_VOICE_AND_DATA -0x03</li> <li>– DMS_DEVICE_CAP_NONSIMUL_VOICE_AND_DATA -0x04</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>

<i>pDevVoiceCaps</i>	<ul style="list-style-type: none"> <li>• Device voice capabilities <ul style="list-style-type: none"> <li>– Bit 0 – GW CSFB <ul style="list-style-type: none"> <li>* 0 – Not capable</li> <li>* 1 – Capable</li> </ul> </li> <li>– Bit 1 – 1x CSFB <ul style="list-style-type: none"> <li>* 0 – Not capable</li> <li>* 1 – Capable</li> </ul> </li> <li>– Bit 2 – VoLTE <ul style="list-style-type: none"> <li>* 0 – Not capable</li> <li>* 1 – Capable</li> </ul> </li> </ul> </li> </ul>
----------------------	--

- Bit to check in ParamPresenceMask - **17**

#### Parameters

<i>pDevVoiceData-Caps</i>	<ul style="list-style-type: none"> <li>• DevVoiceDataCaps <ul style="list-style-type: none"> <li>– Bit 0 – SVLTE capability</li> <li>– Bit 1 – SVDO capability</li> <li>– Bit 2 – SGLTE capability</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pDevMultiSim-Caps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devMultiSimCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pDevMultiSim-VoiceDataCaps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devMultiSimVoiceDataCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pDevCurSubs-Caps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devCurSubsCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pDevSubsVoice-DataCaps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devSubsVoiceDataCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pDevSubs-FeatureMode-Caps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devSubsFeatureModeCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pDevMaxAct-DataSubsCaps</i>	<ul style="list-style-type: none"> <li>• Max number of subscriptions for data activity</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pDevMaxSubs-Caps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devMaxSubsCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>

<i>pDevMaxAct-DataSubsCaps</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_devMaxCfgListCaps</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pDevExplicitCfg-Index</i>	<ul style="list-style-type: none"> <li>• Explicit Cfg Index <ul style="list-style-type: none"> <li>– (-1) - Modem controlled cfg</li> <li>– any other valid value</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.725.2 Field Documentation

8.725.2.1 **dms\_devCaps** unpack\_dms\_GetDeviceCapabilitiesV2\_t::DevCaps

8.725.2.2 **swi\_uint256\_t** unpack\_dms\_GetDeviceCapabilitiesV2\_t::ParamPresenceMask

8.725.2.3 **dms\_devCurSubsCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevCurSubsCaps

8.725.2.4 **int16\_t\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevExplicitCfgIndex

8.725.2.5 **uint8\_t\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevMaxActDataSubsCaps

8.725.2.6 **dms\_devMaxCfgListCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevMaxCfgListCaps

8.725.2.7 **dms\_devMaxSubsCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevMaxSubsCaps

8.725.2.8 **dms\_devMultiSimCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevMultiSimCaps

8.725.2.9 **dms\_devMultiSimVoiceDataCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevMultiSimVoiceDataCaps

8.725.2.10 **uint32\_t\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevSrvCaps

8.725.2.11 **dms\_devSubsFeatureModeCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevSubsFeatureModeCaps

8.725.2.12 **dms\_devSubsVoiceDataCaps\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevSubsVoiceDataCaps

8.725.2.13 **uint64\_t\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevVoiceCaps

8.725.2.14 **uint64\_t\*** unpack\_dms\_GetDeviceCapabilitiesV2\_t::pDevVoiceDataCaps

8.725.2.15 **uint16\_t** unpack\_dms\_GetDeviceCapabilitiesV2\_t::Tlvresult

## 8.726 unpack\_dms\_GetDeviceHardwareRev\_t Struct Reference

## Data Fields

- uint8\_t [stringSize](#)
- char [String](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.726.1 Detailed Description

Structure to store 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>String</i>	<ul style="list-style-type: none"> <li>• NULL terminated Hardware Revision string</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.726.2 Field Documentation

8.726.2.1 [swi\\_uint256\\_t](#) [unpack\\_dms\\_GetDeviceHardwareRev\\_t::ParamPresenceMask](#)

8.726.2.2 [char](#) [unpack\\_dms\\_GetDeviceHardwareRev\\_t::String](#)[255]

8.726.2.3 [uint8\\_t](#) [unpack\\_dms\\_GetDeviceHardwareRev\\_t::stringSize](#)

8.726.2.4 [uint16\\_t](#) [unpack\\_dms\\_GetDeviceHardwareRev\\_t::Tlvresult](#)

## 8.727 [unpack\\_dms\\_GetDeviceMfr\\_t](#) Struct Reference

## Data Fields

- uint8\_t [stringSize](#)
- char [String](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.727.1 Detailed Description

This structure is used to store [unpack\\_dms\\_GetDeviceMfr](#) parameters

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

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>String</i>	<ul style="list-style-type: none"> <li>NULL terminated Device Manufacture string</li> <li>Maximum Length is 255 Bytes</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.727.2 Field Documentation

8.727.2.1 swi\_uint256\_t unpack\_dms\_GetDeviceMfr\_t::ParamPresenceMask

8.727.2.2 char unpack\_dms\_GetDeviceMfr\_t::String[255]

8.727.2.3 uint8\_t unpack\_dms\_GetDeviceMfr\_t::stringSize

8.727.2.4 uint16\_t unpack\_dms\_GetDeviceMfr\_t::Tlvresult

## 8.728 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 [IimeiSvnString](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.728.1 Detailed Description

This structure used to store unpack\_dms\_GetDeviceSerialNumbers parameters

## 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ESNString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>IMEIString</i>	<ul style="list-style-type: none"> <li>• NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>MEIDString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ImeiSvnString</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> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.728.2 Field Documentation



- 8.728.2.1    `uint8_t unpack_dms_GetDeviceSerialNumbers_t::esnSize`
- 8.728.2.2    `char unpack_dms_GetDeviceSerialNumbers_t::ESNString[255]`
- 8.728.2.3    `uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSize`
- 8.728.2.4    `char unpack_dms_GetDeviceSerialNumbers_t::IMEIString[255]`
- 8.728.2.5    `uint8_t unpack_dms_GetDeviceSerialNumbers_t::imeiSvnSize`
- 8.728.2.6    `char unpack_dms_GetDeviceSerialNumbers_t::ImeiSvnString[255]`
- 8.728.2.7    `uint8_t unpack_dms_GetDeviceSerialNumbers_t::meidSize`
- 8.728.2.8    `char unpack_dms_GetDeviceSerialNumbers_t::MEIDString[255]`
- 8.728.2.9    `swi_uint256_t unpack_dms_GetDeviceSerialNumbers_t::ParamPresenceMask`
- 8.728.2.10   `uint16_t unpack_dms_GetDeviceSerialNumbers_t::Tlvresult`

## 8.729    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`
- `swi_uint256_t ParamPresenceMask`

### 8.729.1    Detailed Description

This structure used to store unpack\_dms\_GetFirmwareInfo parameters

#### Parameters

<i>modelid_str</i>	<ul style="list-style-type: none"> <li>- Mode ID String.</li> <li>• NULL-terminated Mode ID String.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>bootversion_str</i>	<ul style="list-style-type: none"> <li>- Boot Version.</li> <li>• NULL-terminated Boot Version String.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>appversion_str</i>	<ul style="list-style-type: none"> <li>- Application Version String.</li> <li>• NULL-terminated Application Version String.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>sku_str</i>	- SKU String. <ul style="list-style-type: none"> <li>• NULL-terminated SKU String.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>packageid_str</i>	- Package ID String. <ul style="list-style-type: none"> <li>• NULL-terminated Package ID String.</li> <li>• deprecated on EM/MC74xx(9x30) devices</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>carrier_str</i>	- Carrier String. <ul style="list-style-type: none"> <li>• NULL-terminated Carrier String.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>priversion_str</i>	- PRI Version String. <ul style="list-style-type: none"> <li>• NULL-terminated PRI Version String.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>cur_carr_name</i>	- Current Carrier Name String. <ul style="list-style-type: none"> <li>• NULL-terminated Current Carrier Name String.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>cur_carr_rev</i>	- Current Carrier Revision String. <ul style="list-style-type: none"> <li>• NULL-terminated Current Carrier Revision String.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.729.2 Field Documentation

8.729.2.1 char unpack\_dms\_GetFirmwareInfo\_t::appversion\_str[85]

8.729.2.2 char unpack\_dms\_GetFirmwareInfo\_t::bootversion\_str[85]

8.729.2.3 char unpack\_dms\_GetFirmwareInfo\_t::carrier\_str[20]

8.729.2.4 char unpack\_dms\_GetFirmwareInfo\_t::cur\_carr\_name[17]

8.729.2.5 char unpack\_dms\_GetFirmwareInfo\_t::cur\_carr\_rev[13]

8.729.2.6 char unpack\_dms\_GetFirmwareInfo\_t::modelid\_str[20]

8.729.2.7 char unpack\_dms\_GetFirmwareInfo\_t::packageid\_str[85]

8.729.2.8 swi\_uint256\_t unpack\_dms\_GetFirmwareInfo\_t::ParamPresenceMask

8.729.2.9 char unpack\_dms\_GetFirmwareInfo\_t::priversion\_str[16]

8.729.2.10 char unpack\_dms\_GetFirmwareInfo\_t::sku\_str[15]

8.729.2.11 uint16\_t unpack\_dms\_GetFirmwareInfo\_t::Tlvresult

## 8.730 unpack\_dms\_GetFirmwareRevision\_t Struct Reference

### Data Fields

- uint8\_t [amssSize](#)
- char [AMSSString](#) [255]
- char [PRIString](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.730.1 Detailed Description

This structure used to store unpack\_dms\_GetFirmwareRevision parameters

#### 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>AMSSString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated AMSS revision string.</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>PRIString[OUT]</i>	<ul style="list-style-type: none"> <li>• NULL-terminated PRI revision string.</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.730.2 Field Documentation

8.730.2.1 uint8\_t unpack\_dms\_GetFirmwareRevision\_t::amssSize

8.730.2.2 char unpack\_dms\_GetFirmwareRevision\_t::AMSSString[255]

8.730.2.3 swi\_uint256\_t unpack\_dms\_GetFirmwareRevision\_t::ParamPresenceMask

8.730.2.4 char unpack\_dms\_GetFirmwareRevision\_t::PRIString[255]

8.730.2.5 uint16\_t unpack\_dms\_GetFirmwareRevision\_t::Tlvresult

## 8.731 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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.731.1 Detailed Description

This structure used to store unpack\_dms\_GetFirmwareRevisions parameters

#### 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>AMSSString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated AMSS revision string</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>BootString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated boot code revision string</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>PRIString</i>	<ul style="list-style-type: none"> <li>• NULL-terminated PRI revision string</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.731.2 Field Documentation

8.731.2.1 `uint8_t unpack_dms_GetFirmwareRevisions_t::amssSize`

8.731.2.2 `char unpack_dms_GetFirmwareRevisions_t::AMSSString[255]`

8.731.2.3 `uint8_t unpack_dms_GetFirmwareRevisions_t::bootSize`

8.731.2.4 `char unpack_dms_GetFirmwareRevisions_t::BootString[255]`

8.731.2.5 `swi_uint256_t unpack_dms_GetFirmwareRevisions_t::ParamPresenceMask`

8.731.2.6 `uint8_t unpack_dms_GetFirmwareRevisions_t::priSize`

8.731.2.7 `char unpack_dms_GetFirmwareRevisions_t::PRIString[255]`

8.731.2.8 `uint16_t unpack_dms_GetFirmwareRevisions_t::Tlvresult`

## 8.732 unpack\_dms\_GetFSN\_t Struct Reference

### Data Fields

- `char String [255]`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.732.1 Detailed Description

This structure used to store Factory Sequence Number parameter

#### Parameters

<i>String</i>	<ul style="list-style-type: none"> <li>• Factory Sequence Number</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.732.2 Field Documentation

8.732.2.1 `swi_uint256_t unpack_dms_GetFSN_t::ParamPresenceMask`

8.732.2.2 `char unpack_dms_GetFSN_t::String[255]`

8.732.2.3 `uint16_t unpack_dms_GetFSN_t::Tlvresult`

## 8.733 unpack\_dms\_GetHardwareRevision\_t Struct Reference

### Data Fields

- char [hwVer](#) [255]
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.733.1 Detailed Description

This structure used to store unpack\_dms\_GetHardwareRevision parameters

#### Parameters

<i>hwVer</i>	<ul style="list-style-type: none"> <li>- Hardware version <ul style="list-style-type: none"> <li>• NULL-terminated string</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul> </li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.733.2 Field Documentation

8.733.2.1 char unpack\_dms\_GetHardwareRevision\_t::hwVer[255]

8.733.2.2 swi\_uint256\_t unpack\_dms\_GetHardwareRevision\_t::ParamPresenceMask

## 8.734 unpack\_dms\_GetIMSI\_t Struct Reference

### Data Fields

- char [imsi](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.734.1 Detailed Description

This structure used to store unpack\_dms\_GetIMSI parameters

#### Parameters

<i>imsi</i>	<ul style="list-style-type: none"> <li>- IMSI no. <ul style="list-style-type: none"> <li>• NULL-terminated String.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>- unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.734.2 Field Documentation

8.734.2.1 char unpack\_dms\_GetIMSI\_t::imsi[255]

8.734.2.2 swi\_uint256\_t unpack\_dms\_GetIMSI\_t::ParamPresenceMask

8.734.2.3 uint16\_t unpack\_dms\_GetIMSI\_t::Tlvresult

## 8.735 unpack\_dms\_GetManufacturer\_t Struct Reference

### Data Fields

- char [manufacturer](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.735.1 Detailed Description

This structure is used to store device manufacturer information.

#### Parameters

<i>manufacturer[O-UT]</i>	<ul style="list-style-type: none"> <li>• NULL terminated string</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.735.2 Field Documentation

8.735.2.1 char unpack\_dms\_GetManufacturer\_t::manufacturer[255]

8.735.2.2 swi\_uint256\_t unpack\_dms\_GetManufacturer\_t::ParamPresenceMask

8.735.2.3 uint16\_t unpack\_dms\_GetManufacturer\_t::Tlvresult

## 8.736 unpack\_dms\_GetModelID\_t Struct Reference

### Data Fields

- char [modelid](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.736.1 Detailed Description

This structure used to store unpack\_dms\_GetModelID parameters

## Parameters

<i>modelid</i>	<ul style="list-style-type: none"> <li>- Device model id.</li> <li>• NULL-terminated String.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.736.2 Field Documentation

8.736.2.1 char unpack\_dms\_GetModelID\_t::modelid[255]

8.736.2.2 swi\_uint256\_t unpack\_dms\_GetModelID\_t::ParamPresenceMask

8.736.2.3 uint16\_t unpack\_dms\_GetModelID\_t::Tlvresult

## 8.737 unpack\_dms\_GetNetworkTime\_t Struct Reference

## Data Fields

- uint16\_t [source](#)
- uint64\_t [timestamp](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.737.1 Detailed Description

This structure used to store unpack\_dms\_GetNetworkTime parameters

## 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> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</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.

## 8.737.2 Field Documentation

8.737.2.1 `swi_uint256_t unpack_dms_GetNetworkTime_t::ParamPresenceMask`

8.737.2.2 `uint16_t unpack_dms_GetNetworkTime_t::source`

8.737.2.3 `uint64_t unpack_dms_GetNetworkTime_t::timestamp`

8.737.2.4 `uint16_t unpack_dms_GetNetworkTime_t::Tlvresult`

## 8.738 unpack\_dms\_GetNetworkTimeV2\_t Struct Reference

## Data Fields

- `uint16_t source`
- `uint64_t timestamp`
- `uint64_t * pSysTime`
- `uint64_t * pUsrTime`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

## 8.738.1 Detailed Description

This structure used to store `unpack_dms_GetNetworkTimeV2` parameters

## Parameters

<i>source</i>	<ul style="list-style-type: none"> <li>• Source of timestamp 0 - 32 kHz device clock 1 - CDMA network 2 - cdma2000 1xEV-DO network</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pSysTime</i>	<ul style="list-style-type: none"> <li>• Count of system time in ms that have elapsed from the start of GPS time (Jan 6, 1980)</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pUsrTime</i>	<ul style="list-style-type: none"> <li>• Count of user time in ms that have elapsed from the start of GPS time (Jan 6, 1980)</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</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.

### 8.738.2 Field Documentation

8.738.2.1 `swi_uint256_t unpack_dms_GetNetworkTimeV2_t::ParamPresenceMask`

8.738.2.2 `uint64_t* unpack_dms_GetNetworkTimeV2_t::pSysTime`

8.738.2.3 `uint64_t* unpack_dms_GetNetworkTimeV2_t::pUsrTime`

8.738.2.4 `uint16_t unpack_dms_GetNetworkTimeV2_t::source`

8.738.2.5 `uint64_t unpack_dms_GetNetworkTimeV2_t::timestamp`

8.738.2.6 `uint16_t unpack_dms_GetNetworkTimeV2_t::Tlvresult`

### 8.739 `unpack_dms_GetOfflineReason_t` Struct Reference

#### Data Fields

- `uint32_t * pReasonMask`
- `uint32_t * pbPlatform`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

#### 8.739.1 Detailed Description

This structure is used to store reason why the operating mode of the device is currently offline.

#### Parameters

<i>pReasonMask[OUT]</i>	<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> <li>Bit to check in ParamPresenceMask - 16</li> </ul>
-------------------------	--

<i>pbPlatform[OUT]</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.739.2 Field Documentation

8.739.2.1 `swi_uint256_t unpack_dms_GetOfflineReason_t::ParamPresenceMask`

8.739.2.2 `uint32_t* unpack_dms_GetOfflineReason_t::pbPlatform`

8.739.2.3 `uint32_t* unpack_dms_GetOfflineReason_t::pReasonMask`

8.739.2.4 `uint16_t unpack_dms_GetOfflineReason_t::Tlvresult`

## 8.740 unpack\_dms\_GetPower\_t Struct Reference

### Data Fields

- `uint32_t` [OperationMode](#)
- `uint32_t` [OfflineReason](#)
- `uint32_t` [HardwareControlledMode](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.740.1 Detailed Description

This structure used to store `unpack_dms_GetPower` parameters

#### Parameters

<i>OperationMode</i>	<ul style="list-style-type: none"> <li>- operating mode. <ul style="list-style-type: none"> <li>• 0 - Online (default).</li> <li>• 1 - Low power (airplane) mode.</li> <li>• 2 - Factory test mode.</li> <li>• 3 - Offline.</li> <li>• 4 - Resetting.</li> <li>• 5 - Power off.</li> <li>• 6 - Persistent low power (airplane) mode.</li> <li>• 7 - Mode - only low power.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul> </li> </ul>
----------------------	---

## Note

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

## Parameters

<i>OfflineReason</i>	- offline reason. <ul style="list-style-type: none"> <li>• 0x0001 - Host image misconfiguration.</li> <li>• 0x0002 - PRI image misconfiguration.</li> <li>• 0x0004 - PRI version incompatible.</li> <li>• 0x0008 - Device memory is full, cannot copy PRI information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Hardware-ControlledMode</i>	- hardware restricted mode. <ul style="list-style-type: none"> <li>• 0x00 - FALSE.</li> <li>• 0x01 - TRUE.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.740.2 Field Documentation

8.740.2.1 `uint32_t unpack_dms_GetPower_t::HardwareControlledMode`

8.740.2.2 `uint32_t unpack_dms_GetPower_t::OfflineReason`

8.740.2.3 `uint32_t unpack_dms_GetPower_t::OperationMode`

8.740.2.4 `swi_uint256_t unpack_dms_GetPower_t::ParamPresenceMask`

8.740.2.5 `uint16_t unpack_dms_GetPower_t::Tlvresult`

8.741 `unpack_dms_GetPRLVersion_t` Struct Reference

## Data Fields

- `uint8_t u8PRLPreference`
- `uint16_t u16PRLVersion`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

## 8.741.1 Detailed Description

This structure used to store `unpack_dms_GetPRLVersion` parameters

## Parameters

<i>u8PRL-Preference</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>u16PRLVersion</i>	<ul style="list-style-type: none"> <li>• PRL version of device.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.741.2 Field Documentation

8.741.2.1 swi\_uint256\_t unpack\_dms\_GetPRLVersion\_t::ParamPresenceMask

8.741.2.2 uint16\_t unpack\_dms\_GetPRLVersion\_t::Tlvresult

8.741.2.3 uint16\_t unpack\_dms\_GetPRLVersion\_t::u16PRLVersion

8.741.2.4 uint8\_t unpack\_dms\_GetPRLVersion\_t::u8PRLPreference

## 8.742 unpack\_dms\_GetSerialNumbers\_t Struct Reference

## Data Fields

- char [esn](#) [255]
- char [imei\\_no](#) [255]
- char [meid](#) [255]
- char [imeisv\\_svn](#) [255]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.742.1 Detailed Description

This structure used to store unpack\_dms\_GetSerialNumbers parameters

## Parameters

<i>esn</i>	<ul style="list-style-type: none"> <li>- Electronic Serial Number of the device. <ul style="list-style-type: none"> <li>• NULL-terminated ESN string. Empty string is returned when ESN is not supported/programmed.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> </ul>
<i>imei_no</i>	<ul style="list-style-type: none"> <li>- International Mobile Equipment Identity of the device. <ul style="list-style-type: none"> <li>• NULL terminated IMEI string. Empty string is returned when IMEI is not supported/programmed.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul> </li> </ul>

<i>meid</i>	<ul style="list-style-type: none"> <li>- Mobile Equipment Identifier of the device.</li> <li>• NULL-terminated MEID string. Empty string is returned when MEID is not supported/programmed.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>imeisv_svn</i>	<ul style="list-style-type: none"> <li>- NULL-terminated IMEI SVN string. Empty string is returned when IMEI SVN is not supported/programmed.imei software version revision.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.742.2 Field Documentation

8.742.2.1 `char unpack_dms_GetSerialNumbers_t::esn[255]`

8.742.2.2 `char unpack_dms_GetSerialNumbers_t::imei_no[255]`

8.742.2.3 `char unpack_dms_GetSerialNumbers_t::imeisv_svn[255]`

8.742.2.4 `char unpack_dms_GetSerialNumbers_t::meid[255]`

8.742.2.5 `swi_uint256_t unpack_dms_GetSerialNumbers_t::ParamPresenceMask`

## 8.743 unpack\_dms\_GetUSBComp\_t Struct Reference

### Data Fields

- `uint8_t USBComp`
- `uint8_t NumSupUSBComps`
- `uint8_t SupUSBComps [255]`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.743.1 Detailed Description

This structure is used to store retrieved USB Composition

## Parameters

<i>USBComp</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 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>NumSupUSB-Comps</i>	<ul style="list-style-type: none"> <li>• Number of supported USB compositions in the parameter to follow</li> <li>• Range - 0-255</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>SupUSBComps</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result.</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.743.2 Field Documentation

8.743.2.1 uint8\_t unpack\_dms\_GetUSBComp\_t::NumSupUSBComps

8.743.2.2 swi\_uint256\_t unpack\_dms\_GetUSBComp\_t::ParamPresenceMask

8.743.2.3 uint8\_t unpack\_dms\_GetUSBComp\_t::SupUSBComps[255]

8.743.2.4 uint16\_t unpack\_dms\_GetUSBComp\_t::Tlvresult

8.743.2.5 uint8\_t unpack\_dms\_GetUSBComp\_t::USBComp

## 8.744 unpack\_dms\_GetVoiceNumber\_t Struct Reference

### Data Fields

- uint8\_t [voiceNumberSize](#)



- char [VoiceNumber](#) [255]
- uint8\_t [minSize](#)
- char [MIN](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.744.1 Detailed Description

Structure to store the voice number in use by the device

#### Parameters

<i>voiceNumber-Size</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that the voice number array can contain.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>VoiceNumber</i>	<ul style="list-style-type: none"> <li>• Voice number string: MDN or MS ISDN</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>MIN</i>	<ul style="list-style-type: none"> <li>• Optional Parameter</li> <li>• MIN string: Empty string returned when MIN is not supported/ programmed.</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.744.2 Field Documentation

8.744.2.1 char unpack\_dms\_GetVoiceNumber\_t::MIN[255]

8.744.2.2 uint8\_t unpack\_dms\_GetVoiceNumber\_t::minSize

8.744.2.3 swi\_uint256\_t unpack\_dms\_GetVoiceNumber\_t::ParamPresenceMask

8.744.2.4 uint16\_t unpack\_dms\_GetVoiceNumber\_t::Tlvresult

8.744.2.5 char unpack\_dms\_GetVoiceNumber\_t::VoiceNumber[255]

8.744.2.6 uint8\_t unpack\_dms\_GetVoiceNumber\_t::voiceNumberSize

## 8.745 unpack\_dms\_PSMCfgChange\_ind\_t Struct Reference

### Data Fields

- [dms\\_PSMEnableStateIndTlv EnableState](#)
- [dms\\_PSMActiveTimerIndTlv ActiveTimer](#)
- [dms\\_PSMPeriodicUpdateTimerIndTlv PeriodicUpdateTimer](#)
- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.745.1 Detailed Description

DMS Event Report indication structure

#### Parameters

<i>EnableState</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMEnableStateIndTlv</a></li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ActiveTimer</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMActiveTimerIndTlv</a></li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>PeriodicUpdate-Timer</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMPeriodicUpdateTimerIndTlv</a></li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.745.2 Field Documentation

8.745.2.1 [dms\\_PSMActiveTimerIndTlv](#) unpack\_dms\_PSMCfgChange\_ind\_t::ActiveTimer

8.745.2.2 [dms\\_PSMEnableStateIndTlv](#) unpack\_dms\_PSMCfgChange\_ind\_t::EnableState

8.745.2.3 [swi\\_uint256\\_t](#) unpack\_dms\_PSMCfgChange\_ind\_t::ParamPresenceMask

8.745.2.4 [dms\\_PSMPeriodicUpdateTimerIndTlv](#) unpack\_dms\_PSMCfgChange\_ind\_t::PeriodicUpdateTimer

8.745.2.5 [uint16\\_t](#) unpack\_dms\_PSMCfgChange\_ind\_t::Tlvresult

## 8.746 unpack\_dms\_ResetToFactoryDefaults\_t Struct Reference

### Data Fields

- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.746.1 Detailed Description

This structure contains reset to factory default unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>

### 8.746.2 Field Documentation

8.746.2.1 `swi_uint256_t unpack_dms_ResetToFactoryDefaults_t::ParamPresenceMask`

8.746.2.2 `uint16_t unpack_dms_ResetToFactoryDefaults_t::Tlvresult`

## 8.747 unpack\_dms\_SetActivationStatusCallback\_t Struct Reference

#### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.747.1 Detailed Description

This structure is used to store Set Service Activation Status callback parameter unpack.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack Result</li></ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>

### 8.747.2 Field Documentation

8.747.2.1 `swi_uint256_t unpack_dms_SetActivationStatusCallback_t::ParamPresenceMask`

8.747.2.2 `uint16_t unpack_dms_SetActivationStatusCallback_t::Tlvresult`

## 8.748 unpack\_dms\_SetCrashAction\_t Struct Reference

#### Data Fields

- `uint8_t notused`
- `swi_uint256_t ParamPresenceMask`

### 8.748.1 Detailed Description

This structure is used to store `unpack_dms_SetCrashAction` parameters

#### Parameters

<i>notused</i>	<ul style="list-style-type: none"> <li>unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.748.2 Field Documentation

8.748.2.1 `uint8_t unpack_dms_SetCrashAction_t::notused`

8.748.2.2 `swi_uint256_t unpack_dms_SetCrashAction_t::ParamPresenceMask`

## 8.749 unpack\_dms\_SetCustFeature\_t Struct Reference

#### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.749.1 Detailed Description

This structure is used to store `unpack_dms_SetCustFeature` parameters

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.749.2 Field Documentation

8.749.2.1 `swi_uint256_t unpack_dms_SetCustFeature_t::ParamPresenceMask`

8.749.2.2 `uint16_t unpack_dms_SetCustFeature_t::Tlvresult`

## 8.750 unpack\_dms\_SetCustFeaturesV2\_t Struct Reference

#### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.750.1 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>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.750.2 Field Documentation

8.750.2.1 `swi_uint256_t unpack_dms_SetCustFeaturesV2_t::ParamPresenceMask`

8.750.2.2 `uint16_t unpack_dms_SetCustFeaturesV2_t::Tlvresult`

## 8.751 unpack\_dms\_SetEventReport\_ind\_t Struct Reference

#### Data Fields

- [dms\\_ActivationStatusTlv](#) ActivationStatusTlv
- [dms\\_OperatingModeTlv](#) OperatingModeTlv
- `uint16_t` Tlvresult
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.751.1 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> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>OperatingMode-Tlv</i>	<ul style="list-style-type: none"> <li>See <a href="#">dms_OperatingModeTlv</a></li> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.751.2 Field Documentation

8.751.2.1 `dms_ActivationStatusTlv unpack_dms_SetEventReport_ind_t::ActivationStatusTlv`

8.751.2.2 **dms\_OperatingModeTlv** `unpack_dms_SetEventReport_ind_t::OperatingModeTlv`

8.751.2.3 **swi\_uint256\_t** `unpack_dms_SetEventReport_ind_t::ParamPresenceMask`

8.751.2.4 **uint16\_t** `unpack_dms_SetEventReport_ind_t::Tlvresult`

## 8.752 `unpack_dms_SetEventReport_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.752.1 Detailed Description

This structure is used to store `unpack_dms_SetEventReport` parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.752.2 Field Documentation

8.752.2.1 **swi\_uint256\_t** `unpack_dms_SetEventReport_t::ParamPresenceMask`

8.752.2.2 **uint16\_t** `unpack_dms_SetEventReport_t::Tlvresult`

## 8.753 `unpack_dms_SetFirmwarePreference_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.753.1 Detailed Description

This structure is used to store `unpack_dms_SetFirmwarePreference` parameters

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.753.2 Field Documentation

8.753.2.1 `swi_uint256_t` `unpack_dms_SetFirmwarePreference_t::ParamPresenceMask`

8.753.2.2 `uint16_t` `unpack_dms_SetFirmwarePreference_t::Tlvresult`

## 8.754 unpack\_dms\_SetIndicationRegister\_t Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.754.1 Detailed Description

This structure is used to store `unpack_dms_SetIndicationRegister` parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.754.2 Field Documentation

8.754.2.1 `swi_uint256_t` `unpack_dms_SetIndicationRegister_t::ParamPresenceMask`

8.754.2.2 `uint16_t` `unpack_dms_SetIndicationRegister_t::Tlvresult`

## 8.755 unpack\_dms\_SetPower\_t Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.755.1 Detailed Description

Structure to store unpack the operating mode of the device.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result.</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.755.2 Field Documentation

8.755.2.1 `swi_uint256_t unpack_dms_SetPower_t::ParamPresenceMask`

8.755.2.2 `uint16_t unpack_dms_SetPower_t::Tlvresult`

## 8.756 `unpack_dms_SetUSBComp_t` Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.756.1 Detailed Description

This structure is used to store unpack USB composition information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv Result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.756.2 Field Documentation

8.756.2.1 `swi_uint256_t unpack_dms_SetUSBComp_t::ParamPresenceMask`

8.756.2.2 `uint16_t unpack_dms_SetUSBComp_t::Tlvresult`

## 8.757 `unpack_dms_SLQSDmsSwiGetPCInfo_t` Struct Reference

### Data Fields

- `uint8_t opMode`
- `int has_LpmFlag`
- `uint32_t LpmFlag`
- `int has_Wdisable`
- `uint8_t Wdisable`
- `int has_PowerOffMode`
- `uint8_t PowerOffMode`
- `int has_PersistentLpm`
- `uint8_t PersistentLpm`
- `swi_uint256_t ParamPresenceMask`

### 8.757.1 Detailed Description

This structure contains the TLV required to Get device power control status information.



## Parameters

<i>opMode</i>	<ul style="list-style-type: none"> <li>Selected operating mode.</li> <li>Values <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>8 - Conducting network test for GSM/WCDMA</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>has_LpmFlag</i>	LPM Flag Availability. <ul style="list-style-type: none"> <li>0 : Unavailable.</li> <li>1 : Available.</li> </ul>

- Bit to check in ParamPresenceMask - **16**

## Parameters

<i>LpmFlag[-Optional]</i>	<ul style="list-style-type: none"> <li>LPM Force Flags</li> <li>Bitmask <ul style="list-style-type: none"> <li>bit0 - Tracks the state of W_DISABLE TLV, indicating state of the W_DISABLE switch.</li> <li>bit1 - Set if LPM is requested by a host request, such as AT command, QMI or MBIM request. Cleared by a host request to return to online mode. Also set when the device is cold or warm booted in persistent LPM.</li> <li>bit2 - Set when the device temperature is outside the valid operating range. Cleared if the temperature returns to the normal range.</li> <li>bit3 - Set when the device voltage is outside the valid operating range. Cleared if the voltage returns to the normal range.</li> <li>bit4 - Set on power up when BIOS locking is enabled. Cleared when the host has disabled the BIOS lock.</li> <li>bit5 - Set if the current device configuration does not match the GOBI image preference.</li> <li>bit6-31 - Additional LPM causes may be added to future products.</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>has_Wdisable</i>	W-Disable Availability. <ul style="list-style-type: none"> <li>0 : Unavailable.</li> <li>1 : Available.</li> </ul>

- Bit to check in ParamPresenceMask - **17**

## Parameters

<i>Wdisable[-Optional]</i>	<ul style="list-style-type: none"> <li>• W_DISABLE</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Switch set to ON position</li> <li>– 1 - Switch set to OFF position</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>has_PowerOff-Mode</i>	Power off mode Availability. <ul style="list-style-type: none"> <li>• 0 : Unavailable.</li> <li>• 1 : Available.</li> </ul>

- Bit to check in ParamPresenceMask - **18**

## Parameters

<i>PowerOffMode[-Optional]</i>	<ul style="list-style-type: none"> <li>• Power-off Mode</li> <li>• Action taken when W_DISABLE is switched to the OFF position <ul style="list-style-type: none"> <li>– 0 - LPM</li> <li>– 1 - Shutdown</li> <li>– 2 - Ignore</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>has_Persistent-Lpm</i>	Persistent LPM Availability. <ul style="list-style-type: none"> <li>• 0 : Unavailable.</li> <li>• 1 : Available.</li> </ul>

- Bit to check in ParamPresenceMask - **19**

## Parameters

<i>PersistentLpm[-Optional]</i>	<ul style="list-style-type: none"> <li>• LPM Persistence</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Non-persistent LPM</li> <li>– 1 - Persistent LPM</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.757.2 Field Documentation

8.757.2.1 int unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::has\_LpmFlag

8.757.2.2 int unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::has\_PersistentLpm

8.757.2.3 int unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::has\_PowerOffMode

8.757.2.4 int unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::has\_Wdisable

8.757.2.5 uint32\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::LpmFlag

8.757.2.6 uint8\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::opMode

8.757.2.7 swi\_uint256\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::ParamPresenceMask

8.757.2.8 uint8\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::PersistentLpm

8.757.2.9 uint8\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::PowerOffMode

8.757.2.10 uint8\_t unpack\_dms\_SLQSDmsSwiGetPCInfo\_t::Wdisable

## 8.758 unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t Struct Reference

### Data Fields

- uint8\_t [type](#)
- uint8\_t [source](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.758.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

#### Parameters

<i>type[OUT]</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>source[OUT]</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 ( 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>

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>Tlvresult[OUT]</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.758.2 Field Documentation

8.758.2.1 `swi_uint256_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::ParamPresenceMask`

8.758.2.2 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::source`

8.758.2.3 `uint16_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::Tlvresult`

8.758.2.4 `uint8_t unpack_dms_SLQSDmsSwiGetResetInfo_Ind_t::type`

8.759 `unpack_dms_SLQSDmsSwiGetResetInfo_t` Struct Reference

## Data Fields

- `uint8_t type`
- `uint8_t source`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

## 8.759.1 Detailed Description

This structure contains the TLV required to Get Reset Info.

## Parameters

<i>type[OUT]</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
------------------	---

<i>source[OUT]</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 ( 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>
--------------------	--

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.759.2 Field Documentation

8.759.2.1 swi\_uint256\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::ParamPresenceMask

8.759.2.2 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::source

8.759.2.3 uint16\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::Tlvresult

8.759.2.4 uint8\_t unpack\_dms\_SLQSDmsSwiGetResetInfo\_t::type

## 8.760 unpack\_dms\_SLQSDmsSwiGetUimSelection\_t Struct Reference

## Data Fields

- uint8\_t uimSelect
- dms\_UimAutoSwitchActSlotTlv \* pUimAutoSwitchActSlot
- swi\_uint256\_t ParamPresenceMask

## 8.760.1 Detailed Description

Structure contains the TLV required to Get currently selected active UIM slot.

## Parameters

<i>uimSelect</i>	<ul style="list-style-type: none"> <li>• Active selected UIM slot</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - slot 1 (e.g. external SIM)</li> <li>– 1 - slot 2 (e.g. embedded SIM)</li> <li>– 2 - remote SIM (if supported)</li> <li>– 3 - SIM auto-switch activated</li> </ul> </li> </ul>
------------------	--

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>pUimAuto-SwitchActSlot</i>	<ul style="list-style-type: none"> <li>• UIM Auto Switch Active slot</li> <li>• see <a href="#">dms_UimAutoSwitchActSlotTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## Note

: None

## 8.760.2 Field Documentation

8.760.2.1 `swi_uint256_t unpack_dms_SLQSDmsSwiGetUimSelection_t::ParamPresenceMask`

8.760.2.2 `dms_UimAutoSwitchActSlotTlv* unpack_dms_SLQSDmsSwiGetUimSelection_t::pUimAutoSwitchActSlot`

8.760.2.3 `uint8_t unpack_dms_SLQSDmsSwiGetUimSelection_t::uimSelect`

8.761 `unpack_dms_SLQSDmsSwiIndicationRegister_t` Struct Reference

## Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

## 8.761.1 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>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.761.2 Field Documentation

8.761.2.1 `swi_uint256_t` `unpack_dms_SLQSDmsSwilIndicationRegister_t::ParamPresenceMask`

8.761.2.2 `uint16_t` `unpack_dms_SLQSDmsSwilIndicationRegister_t::Tlvresult`

## 8.762 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`
- `swi_uint256_t` `ParamPresenceMask`

### 8.762.1 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> <li>• Bit 60 - WCDMA Japan 850 band</li> <li>• Bit 61 - WCDMA 1500 band</li> <li>• Bits 62 and 63 - Reserved</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
----------------------------	--



<i>is_LteBand-Capability_Available</i> [OUT]	<div>LTE Band Capabilities Availability.<ul style="list-style-type: none"><li>• 0 : Unavailable.</li><li>• 1 : Available.</li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul></div>
--	---

<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>• Bit 29 - LTE EUTRAN Band 30 UL: 2350-2360; DL: 2305-2315</li> <li>• Bit 30 - Reserved</li> <li>• Bit 31 - LTE EUTRAN Band 32 DL: 9920-10359</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
--------------------------------	---

<i>is_TdsBand-Capability_Available</i> [OUT]	TDS Band Capabilities Availability. <ul style="list-style-type: none"> <li>• 0 : Unavailable.</li> <li>• 1 : Available.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.762.2 Field Documentation

8.762.2.1 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::bandCapability

8.762.2.2 int unpack\_dms\_SLQSGetBandCapability\_t::is\_LteBandCapability\_Available

8.762.2.3 int unpack\_dms\_SLQSGetBandCapability\_t::is\_TdsBandCapability\_Available

8.762.2.4 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::LteBandCapability

8.762.2.5 swi\_uint256\_t unpack\_dms\_SLQSGetBandCapability\_t::ParamPresenceMask

8.762.2.6 uint64\_t unpack\_dms\_SLQSGetBandCapability\_t::TdsBandCapability

## 8.763 unpack\_dms\_SLQSGetBandCapabilityExt\_t Struct Reference

### Data Fields

- uint64\_t [bandCapability](#)
- int [is\\_LteBandCapability\\_Available](#)
- uint64\_t [LteBandCapability](#)
- int [is\\_TdsBandCapability\\_Available](#)
- uint64\_t [TdsBandCapability](#)
- [dms\\_LteBandsSupport](#) [LteBandsSupport](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.763.1 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> <li>• Bit 60 - WCDMA Japan 850 band</li> <li>• Bit 61 - WCDMA 1500 band</li> <li>• Bits 62 and 63 - Reserved</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
----------------------------	---

<i>is_LteBand-Capability_Available</i> [OUT]	<div>LTE Band Capabilities Availability.<ul style="list-style-type: none"><li>• 0 : Unavailable.</li><li>• 1 : Available.</li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul></div>
--	---

<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>• Bit 29 - LTE EUTRAN Band 30 UL: 2350-2360; DL: 2305-2315</li> <li>• Bit 30 - Reserved</li> <li>• Bit 31 - LTE EUTRAN Band 32 DL: 9920-10359</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
--------------------------------	---

<i>is_TdsBand-Capability_Available[OUT]</i>	TDS Band Capabilities Availability. <ul style="list-style-type: none"> <li>• 0 : Unavailable.</li> <li>• 1 : Available.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>TdsBand-Capability[OUT]</i>	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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>LteBands-Support[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dms_LteBandsSupport</a> for more information</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.763.2 Field Documentation

8.763.2.1 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::bandCapability`

8.763.2.2 `int unpack_dms_SLQSGetBandCapabilityExt_t::is_LteBandCapability_Available`

8.763.2.3 `int unpack_dms_SLQSGetBandCapabilityExt_t::is_TdsBandCapability_Available`

8.763.2.4 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::LteBandCapability`

8.763.2.5 `dms_LteBandsSupport unpack_dms_SLQSGetBandCapabilityExt_t::LteBandsSupport`

8.763.2.6 `swi_uint256_t unpack_dms_SLQSGetBandCapabilityExt_t::ParamPresenceMask`

8.763.2.7 `uint64_t unpack_dms_SLQSGetBandCapabilityExt_t::TdsBandCapability`

## 8.764 unpack\_dms\_SLQSGetERIFile\_t Struct Reference

### Data Fields

- [eriDataparams eriFile](#)
- `uint16_t Tlvresult`
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.764.1 Detailed Description

This structure contains Get ERI file unpack

## Parameters

<i>eriFile</i>	<ul style="list-style-type: none"> <li>• Pointer to structure ERIFileparams</li> <li>• See <a href="#">eriDataparams</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.764.2 Field Documentation

8.764.2.1 [eriDataparams](#) [unpack\\_dms\\_SLQSGetERIFile\\_t::eriFile](#)8.764.2.2 [swi\\_uint256\\_t](#) [unpack\\_dms\\_SLQSGetERIFile\\_t::ParamPresenceMask](#)8.764.2.3 [uint16\\_t](#) [unpack\\_dms\\_SLQSGetERIFile\\_t::Tlvresult](#)8.765 [unpack\\_dms\\_SLQSGetPowerSaveModeConfig\\_t](#) Struct Reference

## Data Fields

- [dms\\_PSMEnableStateTlv](#) \* [pPsmEnableState](#)
- [dms\\_PSMDurationThresholdTlv](#) \* [pDurationThreshold](#)
- [dms\\_PSMDurationDueToOOSTlv](#) \* [pDurationDueToOOS](#)
- [dms\\_PSMRandomizationWindowTlv](#) \* [pRandomizationWindow](#)
- [dms\\_PSMActiveTimerTlv](#) \* [pActiveTimer](#)
- [dms\\_PSMPeriodicUpdateTimerTlv](#) \* [pPeriodicUpdateTimer](#)
- [dms\\_PSMEarlyWakeupTimeTlv](#) \* [pEarlyWakeupTime](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.765.1 Detailed Description

This structure contains the Power Save Mode (PSM) configuration parameter.

## Parameters

<i>pPsmEnableState</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMEnableStateTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pDurationThreshold</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMDurationThresholdTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pDurationDueToOOS</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">dms_PSMDurationDueToOOSTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>



<i>pRandomization-Window</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">dms_PSMRandomizationWindowTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pActiveTimer</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">dms_PSMActiveTimerTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pPeriodic-UpdateTimer</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">dms_PSMPeriodicUpdateTimerTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pEarlyWakeup-Time</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">dms_PSMEarlyWakeupTimeTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.765.2 Field Documentation

8.765.2.1 `dms_PSMActiveTimerTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pActiveTimer`

8.765.2.2 `swi_uint256_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::ParamPresenceMask`

8.765.2.3 `dms_PSMDurationDueToOOSTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pDurationDueToOOS`

8.765.2.4 `dms_PSMDurationThresholdTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pDurationThreshold`

8.765.2.5 `dms_PSMEarlyWakeupTimeTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pEarlyWakeupTime`

8.765.2.6 `dms_PSMPeriodicUpdateTimerTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pPeriodicUpdateTimer`

8.765.2.7 `dms_PSMEnableStateTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pPsmEnableState`

8.765.2.8 `dms_PSMRandomizationWindowTlv*` `unpack_dms_SLQSSetPowerSaveModeConfig_t::pRandomization-Window`

## 8.766 unpack\_dms\_SLQSSetPowerSaveModeConfig\_t Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.766.1 Detailed Description

This structure is used to store `unpack_dms_SLQSSetPowerSaveModeConfig` parameters.

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.766.2 Field Documentation

8.766.2.1 `swi_uint256_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::ParamPresenceMask`8.766.2.2 `uint16_t` `unpack_dms_SLQSSetPowerSaveModeConfig_t::Tlvresult`8.767 `unpack_dms_SLQSSwiClearDyingGaspStatistics_t` Struct Reference

## Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.767.1 Detailed Description

This structure contains Clear Dying GASP unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.767.2 Field Documentation

8.767.2.1 `swi_uint256_t` `unpack_dms_SLQSSwiClearDyingGaspStatistics_t::ParamPresenceMask`8.767.2.2 `uint16_t` `unpack_dms_SLQSSwiClearDyingGaspStatistics_t::Tlvresult`8.768 `unpack_dms_SLQSSwiGetCrashInfo_t` Struct Reference

## Data Fields

- `uint16_t` [Tlvresult](#)
- [crashInfoParams](#) `crashInfoParam`
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.768.1 Detailed Description

This structure contains SWI get crash information unpack information

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>crashInfoParam</i> [- <i>OUT</i> ]	<ul style="list-style-type: none"> <li>• See <a href="#">crashInfoParams</a></li> <li>• crashStatus <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>1</b></li> </ul> </li> <li>• crashInfo <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.768.2 Field Documentation

8.768.2.1 `crashInfoParams unpack_dms_SLQSSwiGetCrashInfo_t::crashInfoParam`8.768.2.2 `swi_uint256_t unpack_dms_SLQSSwiGetCrashInfo_t::ParamPresenceMask`8.768.2.3 `uint16_t unpack_dms_SLQSSwiGetCrashInfo_t::Tlvresult`

## 8.769 unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t Struct Reference

## Data Fields

- [packgetDyingGaspCfg](#) \* [pGetDyingGaspCfg](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.769.1 Detailed Description

This structure contains Get Dying GASP Config unpack

## Parameters

<i>pGetDyingGaspCfg</i>	<ul style="list-style-type: none"> <li>• See <a href="#">packgetDyingGaspCfg</a></li> <li>• pDestSMSNum. <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> <li>• pDestSMSContent. <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>17</b></li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.769.2 Field Documentation

8.769.2.1 `swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::ParamPresenceMask`

8.769.2.2 `packgetDyingGaspCfg* unpack_dms_SLQSSwiGetDyingGaspCfg_t::pGetDyingGaspCfg`

8.769.2.3 `uint16_t unpack_dms_SLQSSwiGetDyingGaspCfg_t::Tlvresult`

## 8.770 unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t Struct Reference

### Data Fields

- [packgetDyingGaspStatistics](#) \* [pGetDyingGaspStatistics](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.770.1 Detailed Description

This structure contains Get Dying GASP Statistics unpack.

#### Parameters

<i>pGetDyingGaspStatistics</i>	<ul style="list-style-type: none"> <li>• See <a href="#">packgetDyingGaspStatistics</a></li> <li>• <code>pTimeStamp</code>. <ul style="list-style-type: none"> <li>– Bit to check in <code>ParamPresenceMask</code> - 16</li> </ul> </li> <li>• <code>pSMSAttemptedFlag</code>. <ul style="list-style-type: none"> <li>– Bit to check in <code>ParamPresenceMask</code> - 17</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>

## 8.770.2 Field Documentation

8.770.2.1 `swi_uint256_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::ParamPresenceMask`

8.770.2.2 `packgetDyingGaspStatistics* unpack_dms_SLQSSwiGetDyingGaspStatistics_t::pGetDyingGaspStatistics`

8.770.2.3 `uint16_t unpack_dms_SLQSSwiGetDyingGaspStatistics_t::Tlvresult`

## 8.771 unpack\_dms\_SLQSSwiGetFirmwareCurr\_t Struct Reference

### Data Fields

- `uint8_t` [numEntries](#)
- `image_info_t` \* [pCurrlmgInfo](#)
- `char` [priver](#) [16]
- `char` [pkgver](#) [16]
- `char` [fwvers](#) [16]
- `char` [carrier](#) [16]
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.771.1 Detailed Description

This structure is used to store unpack\_dms\_SLQSSwiGetFirmwareCurr parameters

#### 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 <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - 1</li> </ul> </li> </ul>
<i>pCurrImgInfo</i> [OUT]	<ul style="list-style-type: none"> <li>Currently Active Image List <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - 1</li> </ul> </li> </ul>
<i>priver</i> [OUT]	<ul style="list-style-type: none"> <li>PRI version of the currently running firmware</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pkgver</i> [OUT]	<ul style="list-style-type: none"> <li>Package version of the currently running firmware</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>fwvers</i> [OUT]	<ul style="list-style-type: none"> <li>firmware version of the currently running firmware</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>carrier</i> [OUT]	<ul style="list-style-type: none"> <li>Carrier string of the currently running firmware</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.771.2 Field Documentation

8.771.2.1 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::carrier[16]

8.771.2.2 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::fwvers[16]

8.771.2.3 uint8\_t unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::numEntries

8.771.2.4 swi\_uint256\_t unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::ParamPresenceMask

8.771.2.5 image\_info\_t\* unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::pCurrImgInfo

8.771.2.6 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::pkgver[16]

8.771.2.7 char unpack\_dms\_SLQSSwiGetFirmwareCurr\_t::priver[16]

## 8.772 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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.772.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>imgType</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Firmware image type that failed the update</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.772.2 Field Documentation

8.772.2.1 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::imgType`

8.772.2.2 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::logString[255]`

8.772.2.3 `swi_uint256_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ParamPresenceMask`

8.772.2.4 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refData`

8.772.2.5 `uint8_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::refString[15]`

8.772.2.6 `uint32_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::ResCode`

8.772.2.7 `uint16_t unpack_dms_SLQSSwiGetFwUpdateStatus_t::Tlvresult`

## 8.773 unpack\_dms\_SLQSSwiGetHostDevInfo\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `char manString [255]`
- `char modelString [255]`
- `char swVerString [255]`
- `char plasmaIDString [255]`
- `char hostID [255]`
- `swi_uint256_t ParamPresenceMask`

### 8.773.1 Detailed Description

This structure contains SWI get host device info unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Result</li> </ul>
------------------	---

<i>manString</i> [OUT]	<ul style="list-style-type: none"> <li>• Host Device Manufacturer Name</li> <li>• Null terminated ASCII String</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>modelString</i> [OUT]	<ul style="list-style-type: none"> <li>• Host Device Model String</li> <li>• Null terminated ASCII String</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>swVerString</i> [OUT]	<ul style="list-style-type: none"> <li>• Host Device Software Version String</li> <li>• Null terminated ASCII string</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>plasmaIDString</i> [OUT]	<ul style="list-style-type: none"> <li>• Host Device Plasma ID String</li> <li>• Null terminated alphanumeric ASCII String.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>hostID</i> [OUT]	<ul style="list-style-type: none"> <li>• Device Host ID String</li> <li>• Null terminated alphanumeric ASCII String.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresenceMask</i> [OUT]	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.773.2 Field Documentation

8.773.2.1 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::hostID[255]

8.773.2.2 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::manString[255]

8.773.2.3 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::modelString[255]

8.773.2.4 swi\_uint256\_t unpack\_dms\_SLQSSwiGetHostDevInfo\_t::ParamPresenceMask

8.773.2.5 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::plasmaIDString[255]

8.773.2.6 char unpack\_dms\_SLQSSwiGetHostDevInfo\_t::swVerString[255]

8.773.2.7 uint16\_t unpack\_dms\_SLQSSwiGetHostDevInfo\_t::Tlvresult

## 8.774 unpack\_dms\_SLQSSwiGetOSInfo\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- char [nameString](#) [255]
- char [versionString](#) [255]



- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.774.1 Detailed Description

This structure contains SWI get host os info unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>nameString[OUT]</i>	<ul style="list-style-type: none"> <li>• Host operating system name</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>VersionString[OUT]</i>	<ul style="list-style-type: none"> <li>• Host operating system version</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.774.2 Field Documentation

8.774.2.1 `char unpack_dms_SLQSSwiGetOSInfo_t::nameString[255]`

8.774.2.2 `swi_uint256_t unpack_dms_SLQSSwiGetOSInfo_t::ParamPresenceMask`

8.774.2.3 `uint16_t unpack_dms_SLQSSwiGetOSInfo_t::Tlvresult`

8.774.2.4 `char unpack_dms_SLQSSwiGetOSInfo_t::versionString[255]`

## 8.775 unpack\_dms\_SLQSSwiGetSerialNoExt\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `char meidString [8]`
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.775.1 Detailed Description

This structure contains SWI get device serial number extension unpack information

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• pack Result</li> </ul>
<i>meidString[OUT]</i>	<ul style="list-style-type: none"> <li>• optional parameter, mobile equipment identifier</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
---------------------------	--

### 8.775.2 Field Documentation

8.775.2.1 `char unpack_dms_SLQSSwiGetSerialNoExt_t::meidString[8]`

8.775.2.2 `swi_uint256_t unpack_dms_SLQSSwiGetSerialNoExt_t::ParamPresenceMask`

8.775.2.3 `uint16_t unpack_dms_SLQSSwiGetSerialNoExt_t::Tlvresult`

## 8.776 `unpack_dms_SLQSSwiSetDyingGaspCfg_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.776.1 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>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.776.2 Field Documentation

8.776.2.1 `swi_uint256_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::ParamPresenceMask`

8.776.2.2 `uint16_t unpack_dms_SLQSSwiSetDyingGaspCfg_t::Tlvresult`

## 8.777 `unpack_dms_SLQSSwiSetHostDevInfo_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.777.1 Detailed Description

This structure contains SWI set host dev info unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.777.2 Field Documentation

8.777.2.1 swi\_uint256\_t unpack\_dms\_SLQSSwiSetHostDevInfo\_t::ParamPresenceMask

8.777.2.2 uint16\_t unpack\_dms\_SLQSSwiSetHostDevInfo\_t::Tlvresult

## 8.778 unpack\_dms\_SLQSSwiSetOSInfo\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.778.1 Detailed Description

This structure contains SWI set host OS info unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.778.2 Field Documentation

8.778.2.1 swi\_uint256\_t unpack\_dms\_SLQSSwiSetOSInfo\_t::ParamPresenceMask

8.778.2.2 uint16\_t unpack\_dms\_SLQSSwiSetOSInfo\_t::Tlvresult

## 8.779 unpack\_dms\_SLQSUIMGetState\_t Struct Reference

## Data Fields

- uint8\_t [state](#)
- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.779.1 Detailed Description

This structure contains UIM get state unpack information

## Parameters

<i>state[OUT]</i>	<ul style="list-style-type: none"> <li>• UIM state</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.779.2 Field Documentation

8.779.2.1 `swi_uint256_t unpack_dms_SLQSUIMGetState_t::ParamPresenceMask`8.779.2.2 `uint8_t unpack_dms_SLQSUIMGetState_t::state`8.779.2.3 `uint16_t unpack_dms_SLQSUIMGetState_t::Tlvresult`8.780 `unpack_dms_SwiEventReportCallBack_ind_t` Struct Reference

## Data Fields

- [dms\\_TemperatureTlv](#) TempTlv
- [dms\\_VoltageTlv](#) VoltTlv
- [dms\\_UimStatusTlv](#) UimStatusTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.780.1 Detailed Description

Structure used to SWI event report indication parameters.

## Parameters

<i>TempTlv</i>	<ul style="list-style-type: none"> <li>• Temperature status tlv</li> <li>• See <a href="#">dms_TemperatureTlv</a> for more information</li> </ul>
<i>VoltTlv</i>	<ul style="list-style-type: none"> <li>• Voltage status tlv</li> <li>• see <a href="#">dms_VoltageTlv</a> for more information</li> </ul>
<i>UimStatusTlv</i>	<ul style="list-style-type: none"> <li>• UIM status tlv</li> <li>• see <a href="#">dms_UimStatusTlv</a> for more information</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

Note

: None

## 8.780.2 Field Documentation

8.780.2.1 swi\_uint256\_t unpack\_dms\_SwiEventReportCallBack\_ind\_t::ParamPresenceMask

8.780.2.2 dms\_TemperatureTlv unpack\_dms\_SwiEventReportCallBack\_ind\_t::TempTlv

8.780.2.3 dms\_UimStatusTlv unpack\_dms\_SwiEventReportCallBack\_ind\_t::UimStatusTlv

8.780.2.4 dms\_VoltageTlv unpack\_dms\_SwiEventReportCallBack\_ind\_t::VoltTlv

## 8.781 unpack\_dms\_SwiSetEventReport\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.781.1 Detailed Description

This structure is used to store unpack\_dms\_SwiSetEventReport parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.781.2 Field Documentation

8.781.2.1 swi\_uint256\_t unpack\_dms\_SwiSetEventReport\_t::ParamPresenceMask

8.781.2.2 uint16\_t unpack\_dms\_SwiSetEventReport\_t::Tlvresult

## 8.782 unpack\_dms\_SwiUimSelect\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.782.1 Detailed Description

This structure is used to store unpack\_dms\_SwiUimSelect parameters.

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack Tlv result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.782.2 Field Documentation

8.782.2.1 `swi_uint256_t unpack_dms_SwiUimSelect_t::ParamPresenceMask`8.782.2.2 `uint16_t unpack_dms_SwiUimSelect_t::Tlvresult`8.783 `unpack_dms_UIMGetControlKeyStatus_t` Struct Reference

## Data Fields

- `uint8_t facilityState`
- `uint8_t verifyRetriesLeft`
- `uint8_t unblockRetriesLeft`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

## 8.783.1 Detailed Description

This structure contains PIN retries status

## Parameters

<i>facilityState[OUT]</i>	<ul style="list-style-type: none"> <li>Control key status <ul style="list-style-type: none"> <li>0 - Deactivated</li> <li>1 - Activated</li> <li>2 - Blocked</li> </ul> </li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>verifyRetriesLeft[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.</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>unblockRetriesLeft[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.</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.783.2 Field Documentation

8.783.2.1 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::facilityState

8.783.2.2 swi\_uint256\_t unpack\_dms\_UIMGetControlKeyStatus\_t::ParamPresenceMask

8.783.2.3 uint16\_t unpack\_dms\_UIMGetControlKeyStatus\_t::Tlvresult

8.783.2.4 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::unlockRetriesLeft

8.783.2.5 uint8\_t unpack\_dms\_UIMGetControlKeyStatus\_t::verifyRetriesLeft

## 8.784 unpack\_dms\_UIMGetICCID\_t Struct Reference

### Data Fields

- uint8\_t [stringSize](#)
- uint8\_t [String](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.784.1 Detailed Description

This structure is used to store unpack\_dms\_UIMGetICCID parameters.

#### Parameters

<i>stringSize</i>	<ul style="list-style-type: none"> <li>• Size of String.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>String</i>	<ul style="list-style-type: none"> <li>• ICCID String.</li> <li>• Maximum Length is 255 Bytes</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unack result.</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.784.2 Field Documentation

8.784.2.1 swi\_uint256\_t unpack\_dms\_UIMGetICCID\_t::ParamPresenceMask

8.784.2.2 uint8\_t unpack\_dms\_UIMGetICCID\_t::String[255]

8.784.2.3 uint8\_t unpack\_dms\_UIMGetICCID\_t::stringSize

8.784.2.4 uint16\_t unpack\_dms\_UIMGetICCID\_t::Tlvresult

## 8.785 unpack\_dms\_UIMGetPINStatus\_t Struct Reference

### Data Fields

- uint8\_t [p1Status](#)
- uint8\_t [p1VerifyRetriesLeft](#)
- uint8\_t [p1UnblockRetriesLeft](#)
- uint8\_t [p2Status](#)
- uint8\_t [p2VerifyRetriesLeft](#)
- uint8\_t [p2UnblockRetriesLeft](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.785.1 Detailed Description

This structure contains Get PIN Status unpack information

#### Parameters

<i>p1Status[OUT]</i>	<ul style="list-style-type: none"> <li>• PIN1 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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>p1VerifyRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of retries left of PIN1, after which PIN will be blocked.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>p1UnblockRetriesLeft[OUT]</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Upon operational failure, this will indicate number of unblock retries left of PIN1, after which the PIN will be permanently blocked i.e. UIM is unusable.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>p2Status[OUT]</i>	<ul style="list-style-type: none"> <li>• PIN2 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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>



<i>p2VerifyRetries-Left[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>Upon operational failure, this will indicate number of retries left of PIN2, after which PIN will be blocked.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>p2Unblock-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 of PIN2, after which the PIN will be permanently blocked i.e. UIM is unusable.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.785.2 Field Documentation

8.785.2.1 `uint8_t unpack_dms_UIMGetPINStatus_t::p1Status`

8.785.2.2 `uint8_t unpack_dms_UIMGetPINStatus_t::p1UnblockRetriesLeft`

8.785.2.3 `uint8_t unpack_dms_UIMGetPINStatus_t::p1VerifyRetriesLeft`

8.785.2.4 `uint8_t unpack_dms_UIMGetPINStatus_t::p2Status`

8.785.2.5 `uint8_t unpack_dms_UIMGetPINStatus_t::p2UnblockRetriesLeft`

8.785.2.6 `uint8_t unpack_dms_UIMGetPINStatus_t::p2VerifyRetriesLeft`

8.785.2.7 `swi_uint256_t unpack_dms_UIMGetPINStatus_t::ParamPresenceMask`

8.785.2.8 `uint16_t unpack_dms_UIMGetPINStatus_t::Tlvresult`

## 8.786 unpack\_dms\_UIMSetControlKeyProtection\_t Struct Reference

### Data Fields

- `uint8_t verifyRetriesLeft`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.786.1 Detailed Description

This structure contains UIM Set control key protection unpack information

## Parameters

<i>verifyRetries-Left[OUT]</i>	<ul style="list-style-type: none"> <li>Optional parameter</li> <li>number of retries left after which the control key is blocked</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.786.2 Field Documentation

8.786.2.1 `swi_uint256_t unpack_dms_UIMSetControlKeyProtection_t::ParamPresenceMask`8.786.2.2 `uint16_t unpack_dms_UIMSetControlKeyProtection_t::Tlvresult`8.786.2.3 `uint8_t unpack_dms_UIMSetControlKeyProtection_t::verifyRetriesLeft`8.787 `unpack_dms_UIMSetPINProtection_t` Struct Reference

## Data Fields

- `uint8_t` [verifyRetriesLeft](#)
- `uint8_t` [unblockRetriesLeft](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.787.1 Detailed Description

This structure contains PIN retries status

## Parameters

<i>verifyRetries-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 PIN will be blocked.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>unblockRetries-Left[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.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult[OUT]</i>	<ul style="list-style-type: none"> <li>Unpack Result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.787.2 Field Documentation

8.787.2.1 swi\_uint256\_t unpack\_dms\_UIMSetPINProtection\_t::ParamPresenceMask

8.787.2.2 uint16\_t unpack\_dms\_UIMSetPINProtection\_t::Tlvresult

8.787.2.3 uint8\_t unpack\_dms\_UIMSetPINProtection\_t::unlockRetriesLeft

8.787.2.4 uint8\_t unpack\_dms\_UIMSetPINProtection\_t::verifyRetriesLeft

## 8.788 unpack\_dms\_UIMUnlockControlKey\_t Struct Reference

### Data Fields

- uint8\_t [unlockRetriesLeft](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.788.1 Detailed Description

This structure contains UIM Set control key protection unpack information

#### Parameters

<i>unlockRetriesLeft</i> [OUT]	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• number of unlock retries left after which the control key is permanently blocked</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i> [OUT]	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.788.2 Field Documentation

8.788.2.1 swi\_uint256\_t unpack\_dms\_UIMUnlockControlKey\_t::ParamPresenceMask

8.788.2.2 uint16\_t unpack\_dms\_UIMUnlockControlKey\_t::Tlvresult

8.788.2.3 uint8\_t unpack\_dms\_UIMUnlockControlKey\_t::unlockRetriesLeft

## 8.789 unpack\_fms\_GetImagesPreference\_t Struct Reference

### Data Fields

- uint32\_t [ImageListSize](#)
- [FMSPrefImageList](#) \* [pImageList](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.789.1 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> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result</li> </ul>

### 8.789.2 Field Documentation

8.789.2.1 `uint32_t unpack_fms_GetImagesPreference_t::ImageListSize`

8.789.2.2 `swi_uint256_t unpack_fms_GetImagesPreference_t::ParamPresenceMask`

8.789.2.3 `FMSPrefImageList* unpack_fms_GetImagesPreference_t::pImageList`

8.789.2.4 `uint16_t unpack_fms_GetImagesPreference_t::Tlvresult`

## 8.790 `unpack_fms_GetStoredImages_t` Struct Reference

### Data Fields

- `uint32_t` [imagelistSize](#)
- [FMSImageList](#) [imageList](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.790.1 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> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>Unpack result</li> </ul>

## 8.790.2 Field Documentation

8.790.2.1 FMSImageList unpack\_fms\_GetStoredImages\_t::imageList

8.790.2.2 uint32\_t unpack\_fms\_GetStoredImages\_t::imagelistSize

8.790.2.3 swi\_uint256\_t unpack\_fms\_GetStoredImages\_t::ParamPresenceMask

8.790.2.4 uint16\_t unpack\_fms\_GetStoredImages\_t::Tlvresult

## 8.791 unpack\_fms\_SetImagesPreference\_t Struct Reference

### Data Fields

- uint32\_t [ImageTypesSize](#)
- uint8\_t [ImageTypes](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.791.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result</li> </ul>

## 8.791.2 Field Documentation

8.791.2.1 uint8\_t unpack\_fms\_SetImagesPreference\_t::ImageTypes[255]

8.791.2.2 uint32\_t unpack\_fms\_SetImagesPreference\_t::ImageTypesSize

8.791.2.3 swi\_uint256\_t unpack\_fms\_SetImagesPreference\_t::ParamPresenceMask

8.791.2.4 uint16\_t unpack\_fms\_SetImagesPreference\_t::Tlvresult

## 8.792 unpack\_ims\_SLQSGetIMSSMSConfig\_t Struct Reference

### Data Fields

- uint8\_t \* [pSettingResp](#)
- uint8\_t \* [pSMSFormat](#)
- uint8\_t \* [pSMSOverIPNwInd](#)

- uint8\_t \* pPhoneCtxtURLen
- uint8\_t \* pPhoneCtxtURI
- swi\_uint256\_t ParamPresenceMask

### 8.792.1 Detailed Description

This structure contains the SLQSGetIMSSMSConfig unpack parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

### 8.792.2 Field Documentation

8.792.2.1 swi\_uint256\_t unpack\_ims\_SLQSGetIMSSMSConfig\_t::ParamPresenceMask

8.792.2.2 uint8\_t\* unpack\_ims\_SLQSGetIMSSMSConfig\_t::pPhoneCtxtURI

8.792.2.3 uint8\_t\* unpack\_ims\_SLQSGetIMSSMSConfig\_t::pPhoneCtxtURLen

8.792.2.4 uint8\_t\* unpack\_ims\_SLQSGetIMSSMSConfig\_t::pSettingResp

8.792.2.5 uint8\_t\* unpack\_ims\_SLQSGetIMSSMSConfig\_t::pSMSFormat

8.792.2.6 uint8\_t\* unpack\_ims\_SLQSGetIMSSMSConfig\_t::pSMSOverIPNwInd

## 8.793 unpack\_ims\_SLQSGetIMSUserConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pSettingResp](#)
- uint8\_t \* [pIMSDomainLen](#)
- uint8\_t \* [pIMSDomain](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.793.1 Detailed Description

This structure contains the SLQSGetIMSUserConfig pack parameters.

## Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pIMSDomainLen</i> [IN/OUT]	<ul style="list-style-type: none"> <li>• Length of IMS Domain Name to follow</li> </ul>
<i>pIMSDomain</i>	<ul style="list-style-type: none"> <li>• IMS domain name</li> <li>• Length of this string must be specified in pIMSDomainLen parameter</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

## 8.793.2 Field Documentation

8.793.2.1 [swi\\_uint256\\_t](#) unpack\_ims\_SLQSGetIMSUserConfig\_t::ParamPresenceMask

8.793.2.2 [uint8\\_t\\*](#) unpack\_ims\_SLQSGetIMSUserConfig\_t::pIMSDomain

8.793.2.3 [uint8\\_t\\*](#) unpack\_ims\_SLQSGetIMSUserConfig\_t::pIMSDomainLen

8.793.2.4 [uint8\\_t\\*](#) unpack\_ims\_SLQSGetIMSUserConfig\_t::pSettingResp

## 8.794 unpack\_ims\_SLQSGetIMSVolIPConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pSettingResp](#)
- uint16\_t \* [pSessionExpiryTimer](#)
- uint16\_t \* [pMinSessionExpiryTimer](#)
- uint8\_t \* [pAmrWbEnable](#)
- uint8\_t \* [pScrAmrEnable](#)
- uint8\_t \* [pScrAmrWbEnable](#)
- uint8\_t \* [pAmrMode](#)
- uint16\_t \* [pAmrWBMode](#)
- uint8\_t \* [pAmrOctetAligned](#)
- uint8\_t \* [pAmrWBOctetAligned](#)
- uint16\_t \* [pRingingTimer](#)
- uint16\_t \* [pRingBackTimer](#)
- uint16\_t \* [pRTPRTCPInactTimer](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.794.1 Detailed Description

This structure contains the SLQSGetIMSVoIPConfig unpack 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pSessionExpiry-Timer</i>	<ul style="list-style-type: none"> <li>Session duration, in seconds</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pMinSession-ExpiryTimer</i>	<ul style="list-style-type: none"> <li>Minimum allowed value for session expiry timer, in seconds</li> <li>Bit to check in ParamPresenceMask - <b>18</b></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> <li>Bit to check in ParamPresenceMask - <b>19</b></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> <li>Bit to check in ParamPresenceMask - <b>20</b></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> <li>Bit to check in ParamPresenceMask - <b>21</b></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> <li>Bit to check in ParamPresenceMask - <b>22</b></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> <li>• Bit to check in ParamPresenceMask - <b>23</b></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> <li>• Bit to check in ParamPresenceMask - <b>24</b></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> <li>• Bit to check in ParamPresenceMask - <b>25</b></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> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></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> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>

## 8.794.2 Field Documentation

8.794.2.1 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pAmrMode

8.794.2.2 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pAmrOctetAligned

8.794.2.3 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pAmrWbEnable

- 8.794.2.4 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pAmrWBMode
- 8.794.2.5 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pAmrWBOctetAligned
- 8.794.2.6 swi\_uint256\_t unpack\_ims\_SLQSGetIMSVolPConfig\_t::ParamPresenceMask
- 8.794.2.7 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pMinSessionExpiryTimer
- 8.794.2.8 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pRingBackTimer
- 8.794.2.9 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pRingingTimer
- 8.794.2.10 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pRTPRTCPInactTimer
- 8.794.2.11 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pScrAmrEnable
- 8.794.2.12 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pScrAmrWbEnable
- 8.794.2.13 uint16\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pSessionExpiryTimer
- 8.794.2.14 uint8\_t\* unpack\_ims\_SLQSGetIMSVolPConfig\_t::pSettingResp

## 8.795 unpack\_ims\_SLQSGetRegMgrConfig\_t Struct Reference

### Data Fields

- uint8\_t \* [pSettingResp](#)
- uint16\_t \* [pPCSCFPort](#)
- uint8\_t \* [pPriCSCFPortNameLen](#)
- uint8\_t \* [pPriCSCFPortName](#)
- uint8\_t \* [pIMSTestMode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.795.1 Detailed Description

This structure contains the SLQSGetRegMgrConfig unpack parameters.

#### Parameters

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>• Settings Response</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPCSCFPort</i>	<ul style="list-style-type: none"> <li>• Proxy call session control function port</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pPriCSCFPort-NameLen(IN/OUT)</i>	<ul style="list-style-type: none"> <li>• Size in bytes assigned to the primary CSCF Port 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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
---------------------	---

**Note**

pPriCSCFPortNameLen must be set to a valid value during API call to retrieve pPriCSCFPortName.

**8.795.2 Field Documentation**

8.795.2.1 swi\_uint256\_t unpack\_ims\_SLQSGetRegMgrConfig\_t::ParamPresenceMask

8.795.2.2 uint8\_t\* unpack\_ims\_SLQSGetRegMgrConfig\_t::pIMSTestMode

8.795.2.3 uint16\_t\* unpack\_ims\_SLQSGetRegMgrConfig\_t::pPCSCFPort

8.795.2.4 uint8\_t\* unpack\_ims\_SLQSGetRegMgrConfig\_t::pPriCSCFPortName

8.795.2.5 uint8\_t\* unpack\_ims\_SLQSGetRegMgrConfig\_t::pPriCSCFPortNameLen

8.795.2.6 uint8\_t\* unpack\_ims\_SLQSGetRegMgrConfig\_t::pSettingResp

**8.796 unpack\_ims\_SLQSGetSIPConfig\_t Struct Reference****Data Fields**

- uint8\_t \* [pSettingResp](#)
- uint16\_t \* [pSIPLocalPort](#)
- uint32\_t \* [pTimerSIPReg](#)
- uint32\_t \* [pSubscribeTimer](#)
- uint32\_t \* [pTimerT1](#)
- uint32\_t \* [pTimerT2](#)
- uint32\_t \* [pTimerTf](#)
- uint8\_t \* [pSigCompEnabled](#)
- swi\_uint256\_t [ParamPresenceMask](#)

**8.796.1 Detailed Description**

This structure contains the SLQSGetSIPConfig unpack parameters.

**Parameters**

<i>pSettingResp</i>	<ul style="list-style-type: none"> <li>Settings Response</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pSIPLocalPort</i>	<ul style="list-style-type: none"> <li>Primary call session control function SIP port number</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>pTimerSIPReg</i>	<ul style="list-style-type: none"> <li>Initial SIP registration duration from the User equipment, in seconds</li> <li>Bit to check in ParamPresenceMask - <b>18</b></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> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pTimerT1</i>	<ul style="list-style-type: none"> <li>RTT estimate, in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>20</b></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> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pTimerTf</i>	<ul style="list-style-type: none"> <li>Non-invite transaction timeout timer, in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>22</b></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> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>

## 8.796.2 Field Documentation

8.796.2.1 `swi_uint256_t unpack_ims_SLQSGetSIPConfig_t::ParamPresenceMask`

8.796.2.2 `uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSettingResp`

8.796.2.3 `uint8_t* unpack_ims_SLQSGetSIPConfig_t::pSigCompEnabled`

8.796.2.4 `uint16_t* unpack_ims_SLQSGetSIPConfig_t::pSIPLocalPort`

8.796.2.5 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pSubscribeTimer`

8.796.2.6 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerSIPReg`

8.796.2.7 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT1`

8.796.2.8 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerT2`

8.796.2.9 `uint32_t* unpack_ims_SLQSGetSIPConfig_t::pTimerTf`

## 8.797 `unpack_ims_SLQSRegMgrCfgCallBack_ind_t` Struct Reference

### Data Fields

- [ims\\_PCSCFPortInfo PCTlv](#)

- [ims\\_CSCFPortNameInfo](#) PNTlv
- [ims\\_IMSTestModelInfo](#) TMTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.797.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

#### Parameters

<i>PCTlv</i>	<ul style="list-style-type: none"> <li>• Primary PCSCF Port info</li> <li>• See <a href="#">ims_PCSCFPortInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>PNTlv</i>	<ul style="list-style-type: none"> <li>• CSCF Port info</li> <li>• see <a href="#">ims_CSCFPortNameInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>TMTlv</i>	<ul style="list-style-type: none"> <li>• IMS test mode info</li> <li>• see <a href="#">ims_IMSTestModelInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

#### Note

: None

### 8.797.2 Field Documentation

8.797.2.1 [swi\\_uint256\\_t](#) unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t::ParamPresenceMask

8.797.2.2 [ims\\_PCSCFPortInfo](#) unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t::PCTlv

8.797.2.3 [ims\\_CSCFPortNameInfo](#) unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t::PNTlv

8.797.2.4 [ims\\_IMSTestModelInfo](#) unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t::TMTlv

## 8.798 unpack\_ims\_SLQSSetIMSSMSConfig\_t Struct Reference

#### Data Fields

- [uint8\\_t](#) \* [pSettingResp](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.798.1 Detailed Description

This structure contains the SLQSSetIMSSMSConfig unpack 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
---------------------	--

## 8.798.2 Field Documentation

8.798.2.1 `swi_uint256_t` `unpack_ims_SLQSSetIMSSMSConfig_t::ParamPresenceMask`8.798.2.2 `uint8_t*` `unpack_ims_SLQSSetIMSSMSConfig_t::pSettingResp`8.799 `unpack_ims_SLQSSetIMSUserConfig_t` Struct Reference

## Data Fields

- `uint8_t *` [pSettingResp](#)
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

## 8.799.1 Detailed Description

This structure contains the SLQSSetIMSUserConfig unpack 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
---------------------	--

## 8.799.2 Field Documentation

8.799.2.1 `swi_uint256_t` `unpack_ims_SLQSSetIMSUserConfig_t::ParamPresenceMask`8.799.2.2 `uint8_t*` `unpack_ims_SLQSSetIMSUserConfig_t::pSettingResp`8.800 `unpack_ims_SLQSSetIMSVoIPConfig_t` Struct Reference

## Data Fields

- `uint8_t *` [pSettingResp](#)
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

## 8.800.1 Detailed Description

This structure contains the SLQSSetIMSVoIPConfig unpack 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><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>
---------------------	---

## 8.800.2 Field Documentation

8.800.2.1 swi\_uint256\_t unpack\_ims\_SLQSSetIMSVolPConfig\_t::ParamPresenceMask

8.800.2.2 uint8\_t\* unpack\_ims\_SLQSSetIMSVolPConfig\_t::pSettingResp

## 8.801 unpack\_ims\_SLQSSetRegMgrConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pSettingResp](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.801.1 Detailed Description

This structure contains the SLQSSetRegMgrConfig unpack 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><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>
---------------------	---

## 8.801.2 Field Documentation

8.801.2.1 swi\_uint256\_t unpack\_ims\_SLQSSetRegMgrConfig\_t::ParamPresenceMask

8.801.2.2 uint8\_t\* unpack\_ims\_SLQSSetRegMgrConfig\_t::pSettingResp

## 8.802 unpack\_ims\_SLQSSetSIPConfig\_t Struct Reference

## Data Fields

- uint8\_t \* [pSettingResp](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.802.1 Detailed Description

This structure contains the SLQSSetSIPConfig unpack 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
---------------------	--

## 8.802.2 Field Documentation

8.802.2.1 swi\_uint256\_t unpack\_ims\_SLQSSetSIPConfig\_t::ParamPresenceMask

8.802.2.2 uint8\_t\* unpack\_ims\_SLQSSetSIPConfig\_t::pSettingResp

## 8.803 unpack\_ims\_SLQSSIPCfgCallback\_ind\_t Struct Reference

## Data Fields

- [ims\\_SIPPortInfo](#) SPTlv
- [ims\\_SIPRegnTmrInfo](#) SRTlv
- [ims\\_SubscrTmrInfo](#) STTlv
- [ims\\_TmrT1Info](#) TT1Tlv
- [ims\\_TmrT2Info](#) TT2Tlv
- [ims\\_TmrTfInfo](#) TTfTlv
- [ims\\_SigCompEnInfo](#) SCTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.803.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

## Parameters

<i>SPTlv</i>	<ul style="list-style-type: none"> <li>SIP port tlv</li> <li>See <a href="#">ims_SIPPortInfo</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>SRTlv</i>	<ul style="list-style-type: none"> <li>SIP Registration Timer tlv</li> <li>see <a href="#">ims_SIPRegnTmrInfo</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>STTlv</i>	<ul style="list-style-type: none"> <li>Subscriber timer tlv</li> <li>see <a href="#">ims_SubscrTmrInfo</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>TT1Tlv</i>	<ul style="list-style-type: none"> <li>Timer T1 tlv</li> <li>see <a href="#">ims_TmrT1Info</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>



<i>TT2Tlv</i>	<ul style="list-style-type: none"> <li>• Timer T2 tlv</li> <li>• see <a href="#">ims_TmrT2Info</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>TTfTlv</i>	<ul style="list-style-type: none"> <li>• Timer Tf tlv</li> <li>• see <a href="#">ims_TmrTfInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>SCTlv</i>	<ul style="list-style-type: none"> <li>• SigComp Status Tlv</li> <li>• see <a href="#">ims_SigCompEnInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>

**Note**

: None

**8.803.2 Field Documentation**8.803.2.1 `swi_uint256_t unpack_ims_SLQSSIPCfgCallBack_ind_t::ParamPresenceMask`8.803.2.2 `ims_SigCompEnInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SCTlv`8.803.2.3 `ims_SIPPortInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SPTlv`8.803.2.4 `ims_SIPRegnTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::SRTlv`8.803.2.5 `ims_SubscrTmrInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::STTlv`8.803.2.6 `ims_TmrT1Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT1Tlv`8.803.2.7 `ims_TmrT2Info unpack_ims_SLQSSIPCfgCallBack_ind_t::TT2Tlv`8.803.2.8 `ims_TmrTfInfo unpack_ims_SLQSSIPCfgCallBack_ind_t::TTfTlv`**8.804 unpack\_ims\_SLQSSMSCfgCallBack\_ind\_t Struct Reference****Data Fields**

- [ims\\_SMSFmtInfo](#) SFTlv
- [ims\\_SMSolPNwInfo](#) SINTlv
- [ims\\_PhCtxtURIInfo](#) PCURTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

**8.804.1 Detailed Description**

Structure used to store SMS Config Indication Parameters.

## Parameters

<i>SFTIv</i>	<ul style="list-style-type: none"> <li>• SMS format info</li> <li>• See <a href="#">ims_SMSFmtlInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>SINTIv</i>	<ul style="list-style-type: none"> <li>• SMS over IP network info</li> <li>• see <a href="#">ims_SMSoIPNwInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>PCURTIv</i>	<ul style="list-style-type: none"> <li>• Phone context URI info</li> <li>• see <a href="#">ims_PhCtxtURIInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

## Note

: None

## 8.804.2 Field Documentation

8.804.2.1 `swi_uint256_t` `unpack_ims_SLQSSMSCfgCallBack_ind_t::ParamPresenceMask`

8.804.2.2 `ims_PhCtxtURIInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::PCURTIv`

8.804.2.3 `ims_SMSFmtlInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::SFTIv`

8.804.2.4 `ims_SMSoIPNwInfo` `unpack_ims_SLQSSMSCfgCallBack_ind_t::SINTIv`

8.805 `unpack_ims_SLQSUUserCfgCallBack_ind_t` Struct Reference

## Data Fields

- [ims\\_IMSDomainInfo](#) `IDTIv`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

## 8.805.1 Detailed Description

Structure used to store SMS Config Indication Parameters.

## Parameters

<i>IDTIv</i>	<ul style="list-style-type: none"> <li>• IMS domain info</li> <li>• See <a href="#">ims_IMSDomainInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
--------------	--

Note

: None

## 8.805.2 Field Documentation

8.805.2.1 `ims_IMSDomainInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::IDTlv`

8.805.2.2 `swi_uint256_t` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ParamPresenceMask`

## 8.806 unpack\_ims\_SLQSVolPCfgCallBack\_ind\_t Struct Reference

### Data Fields

- [ims\\_SessDurInfo](#) SDTlv
- [ims\\_MinSessExpInfo](#) MSETlv
- [ims\\_EnabAMRWBInfo](#) EAWTlv
- [ims\\_EnabSCRAMRInfo](#) ESATlv
- [ims\\_EnabSCRAMRWBInfo](#) ESAWTlv
- [ims\\_AMRModelInfo](#) AMTlv
- [ims\\_AMRWBModelInfo](#) AWMTlv
- [ims\\_AMROctAlgnInfo](#) AOATlv
- [ims\\_AMRWBOctAlgnInfo](#) AWOATlv
- [ims\\_RngTmrInfo](#) RTTlv
- [ims\\_RngBkTmrInfo](#) RBTTlv
- [ims\\_RTPRTCPInactTmrDurInfo](#) RTIDTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.806.1 Detailed Description

Structure used to store VOIP Config Indication Parameters.

#### Parameters

<i>SDTlv</i>	<ul style="list-style-type: none"> <li>• Session Duration tlv</li> <li>• See <a href="#">ims_SessDurInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>MSETlv</i>	<ul style="list-style-type: none"> <li>• Minimum Session Timer tlv</li> <li>• see <a href="#">ims_MinSessExpInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>EAWTlv</i>	<ul style="list-style-type: none"> <li>• Enable AMR WB tlv</li> <li>• see <a href="#">ims_EnabAMRWBInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ESATlv</i>	<ul style="list-style-type: none"> <li>• Enable SCR AMR NB tlv</li> <li>• see <a href="#">ims_EnabSCRAMRInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>ESAWTlv</i>	<ul style="list-style-type: none"> <li>• Enable SCR AMR WB tlv</li> <li>• see <a href="#">ims_EnabSCRAMRWBInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>AMTlv</i>	<ul style="list-style-type: none"> <li>• AMR NB Mode tlv</li> <li>• see <a href="#">ims_AMRModelInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>AWMTlv</i>	<ul style="list-style-type: none"> <li>• AMR WB Mode Tlv</li> <li>• see <a href="#">ims_AMRWBModelInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>AOATlv</i>	<ul style="list-style-type: none"> <li>• AMR NB Octet Aligned tlv</li> <li>• see <a href="#">ims_AMROctAlgnInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>AWOATlv</i>	<ul style="list-style-type: none"> <li>• AMR WB Octet Aligned tlv</li> <li>• see <a href="#">ims_AMRWBOctAlgnInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>RTTlv</i>	<ul style="list-style-type: none"> <li>• Ringing Timer Duration tlv</li> <li>• see <a href="#">ims_RngTmrInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>RBTTlv</i>	<ul style="list-style-type: none"> <li>• Ringback Duration tlv</li> <li>• see <a href="#">ims_RngBkTmrInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>RTIDTlv</i>	<ul style="list-style-type: none"> <li>• RTP/RTCP Inactivity Timer Duration Tlv</li> <li>• see <a href="#">ims_RTPRTCPInactTmrDurInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>

**Note**

: None

**8.806.2 Field Documentation**8.806.2.1 **ims\_AMRModelInfo** unpack\_ims\_SLQSVolPCfgCallBack\_ind\_t::AMTlv8.806.2.2 **ims\_AMROctAlgnInfo** unpack\_ims\_SLQSVolPCfgCallBack\_ind\_t::AOATlv8.806.2.3 **ims\_AMRWBModelInfo** unpack\_ims\_SLQSVolPCfgCallBack\_ind\_t::AWMTlv

- 8.806.2.4 `ims_AMRWBOctAlgnInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::AWOATlv`
- 8.806.2.5 `ims_EnabAMRWBInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::EAWTlv`
- 8.806.2.6 `ims_EnabSCRAMRInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ESATlv`
- 8.806.2.7 `ims_EnabSCRAMRWBInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ESAWTlv`
- 8.806.2.8 `ims_MinSessExpInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::MSETlv`
- 8.806.2.9 `swi_uint256_t` `unpack_ims_SLQSVolPCfgCallBack_ind_t::ParamPresenceMask`
- 8.806.2.10 `ims_RngBkTmrInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RBTTlv`
- 8.806.2.11 `ims_RTPRTCPlnactTmrDurInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RTIDTlv`
- 8.806.2.12 `ims_RngTmrInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::RTTlv`
- 8.806.2.13 `ims_SessDurInfo` `unpack_ims_SLQSVolPCfgCallBack_ind_t::SDTlv`

## 8.807 unpack\_imsa\_SLQSGetIMSARegStatus\_t Struct Reference

### Data Fields

- `uint8_t` [ImsRegStatus](#)
- `uint16_t` [ImsRegErrCode](#)
- `uint32_t` [NewImsRegStatus](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.807.1 Detailed Description

This structure contains response parameters of registration status.

#### Parameters

<i>ImsRegStatus</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ImsRegErrCode</i>	<ul style="list-style-type: none"> <li>• IMS Registration Error Code.</li> <li>• An error code is returned when the IMS registration status is IMSA_STATUS_NOT_REGISTERED. -Values <ul style="list-style-type: none"> <li>– 3xx - Redirection responses</li> <li>– 4xx - Client failure responses</li> <li>– 5xx - Server failure responses</li> <li>– 6xx - Global failure responses</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>NewImsReg-Status</i>	<ul style="list-style-type: none"> <li>• New IMS Registration Status</li> <li>• Values <ul style="list-style-type: none"> <li>– 0 - Not registered for IMS</li> <li>– 1 - Registering for IMS</li> <li>– 2 - Registered for IMS</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
-------------------------	---

## 8.807.2 Field Documentation

8.807.2.1 uint16\_t unpack\_imsa\_SLQSGetIMSARegStatus\_t::ImsRegErrCode

8.807.2.2 uint8\_t unpack\_imsa\_SLQSGetIMSARegStatus\_t::ImsRegStatus

8.807.2.3 uint32\_t unpack\_imsa\_SLQSGetIMSARegStatus\_t::NewImsRegStatus

8.807.2.4 swi\_uint256\_t unpack\_imsa\_SLQSGetIMSARegStatus\_t::ParamPresenceMask

## 8.808 unpack\_imsa\_SLQSGetIMSAServiceStatus\_t Struct Reference

### Data Fields

- uint32\_t [SmsServiceStatus](#)
- uint32\_t [VoipServiceStatus](#)
- uint32\_t [VtServiceStatus](#)
- uint32\_t [SmsServiceRat](#)
- uint32\_t [VoipServiceRat](#)
- uint32\_t [VtServiceRat](#)
- uint32\_t [UtServiceStatus](#)
- uint32\_t [UtServiceRat](#)
- uint32\_t [VsServiceStatus](#)
- uint32\_t [VsServiceRat](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.808.1 Detailed Description

This structure contains response parameters of service status for various IMS services.

#### Parameters

<i>SmsService-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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
--------------------------	---

<i>VoipServiceStatus</i>	<ul style="list-style-type: none"> <li>• VoIP Service Status.</li> <li>• 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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>VtServiceStatus</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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>SmsServiceRat</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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>VoipServiceRat</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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>VtServiceRat</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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>UtServiceStatus</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> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>

<i>UtServiceRat</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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>VsServiceStatus</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> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>VsServiceRat</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> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>

## 8.808.2 Field Documentation

8.808.2.1 `swi_uint256_t unpack_imsa_SLQSGetIMSAServiceStatus_t::ParamPresenceMask`

8.808.2.2 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceRat`

8.808.2.3 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::SmsServiceStatus`

8.808.2.4 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceRat`

8.808.2.5 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::UtServiceStatus`

8.808.2.6 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceRat`

8.808.2.7 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VoipServiceStatus`

8.808.2.8 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceRat`

8.808.2.9 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VsServiceStatus`

8.808.2.10 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceRat`

8.808.2.11 `uint32_t unpack_imsa_SLQSGetIMSAServiceStatus_t::VtServiceStatus`

## 8.809 `unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t` Struct Reference



## Data Fields

- uint8\_t [PdpConnState](#)
- [imsa\\_IMSFailErrCodeTlv](#) FailErrCode
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.809.1 Detailed Description

Structure used to store IMSA PDP status indication Parameters.

#### Parameters

<i>PdpConnState</i>	<ul style="list-style-type: none"><li>• Values:<ul style="list-style-type: none"><li>– 0- not connected</li><li>– 1- connected</li></ul></li><li>• Bit to check in ParamPresenceMask - <b>1</b></li></ul>
<i>FailErrCode</i>	<ul style="list-style-type: none"><li>• Fail error code when PDP is not connected</li><li>• See <a href="#">imsa_IMSFailErrCodeTlv</a> for more information</li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>

#### Note

: None

### 8.809.2 Field Documentation

8.809.2.1 [imsa\\_IMSFailErrCodeTlv](#) unpack\_imsa\_SLQSImsaPdpStatusCallBack\_ind\_t::FailErrCode

8.809.2.2 [swi\\_uint256\\_t](#) unpack\_imsa\_SLQSImsaPdpStatusCallBack\_ind\_t::ParamPresenceMask

8.809.2.3 [uint8\\_t](#) unpack\_imsa\_SLQSImsaPdpStatusCallBack\_ind\_t::PdpConnState

## 8.810 unpack\_imsa\_SLQSImsaRatStatusCallBack\_ind\_t Struct Reference

## Data Fields

- [imsa\\_RatHandoverStatusInfo](#) RatHandover
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.810.1 Detailed Description

Structure used to store IMSA RAT status indication Parameters.

#### Parameters

<i>RatHandover</i>	<ul style="list-style-type: none"><li>• RAT handover status info</li><li>• See <a href="#">imsa_RatHandoverStatusInfo</a> for more information</li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>
--------------------	--

## Note

: None

## 8.810.2 Field Documentation

8.810.2.1 swi\_uint256\_t unpack\_imsa\_SLQSImsaRatStatusCallBack\_ind\_t::ParamPresenceMask

8.810.2.2 imsa\_RatHandoverStatusInfo unpack\_imsa\_SLQSImsaRatStatusCallBack\_ind\_t::RatHandover

## 8.811 unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t Struct Reference

## Data Fields

- [imsa\\_IMSRegStatusInfo](#) IMSRegistration
- [imsa\\_IMSRegStatusCodeInfo](#) IMSRegistrationError
- [imsa\\_NewIMSRegStatusInfo](#) NewIMSRegistration
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.811.1 Detailed Description

Structure used to store IMSA registration status indication Parameters.

## Parameters

<i>IMSRegistration</i>	<ul style="list-style-type: none"> <li>• IMS registration info</li> <li>• See <a href="#">imsa_IMSRegStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>IMSRegistration-Error</i>	<ul style="list-style-type: none"> <li>• IMS registration error</li> <li>• see <a href="#">imsa_IMSRegStatusCodeInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>NewIMS-Registration</i>	<ul style="list-style-type: none"> <li>• New IMS registration info</li> <li>• see <a href="#">imsa_NewIMSRegStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

## Note

: None

## 8.811.2 Field Documentation

8.811.2.1 imsa\_IMSRegStatusInfo unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t::IMSRegistration

8.811.2.2 imsa\_IMSRegStatusCodeInfo unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t::IMSRegistrationError

8.811.2.3 imsa\_NewIMSRegStatusInfo unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t::NewIMSRegistration

8.811.2.4 swi\_uint256\_t unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t::ParamPresenceMask

## 8.812 unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t Struct Reference

### Data Fields

- [imsa\\_SmsSvcStatusInfo](#) SmsService
- [imsa\\_VoipSvcStatusInfo](#) VoipService
- [imsa\\_VtSvcStatusInfo](#) VtService
- [imsa\\_SmsRatInfo](#) SmsRat
- [imsa\\_VoipRatInfo](#) VoipRat
- [imsa\\_VtRatInfo](#) VtRat
- [imsa\\_UtSvcStatusInfo](#) UtService
- [imsa\\_UtRatInfo](#) UtRat
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.812.1 Detailed Description

Structure used to store IMSA service status indication Parameters.

#### Parameters

<i>SmsService</i>	<ul style="list-style-type: none"> <li>• SMS service info</li> <li>• See <a href="#">imsa_SmsSvcStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>VoipService</i>	<ul style="list-style-type: none"> <li>• VOIP service info</li> <li>• see <a href="#">imsa_VoipSvcStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>VtService</i>	<ul style="list-style-type: none"> <li>• VT service info</li> <li>• see <a href="#">imsa_VtSvcStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>SmsRat</i>	<ul style="list-style-type: none"> <li>• SMS RAT info</li> <li>• see <a href="#">imsa_SmsRatInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>VoipRat</i>	<ul style="list-style-type: none"> <li>• VOIP RAT info</li> <li>• see <a href="#">imsa_VoipRatInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>VtRat</i>	<ul style="list-style-type: none"> <li>• VT RAT info</li> <li>• see <a href="#">imsa_VtRatInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>

<i>UtService</i>	<ul style="list-style-type: none"> <li>• UT service info</li> <li>• see <a href="#">imsa_UtSvcStatusInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>UtRat</i>	<ul style="list-style-type: none"> <li>• UT RAT info</li> <li>• see <a href="#">imsa_UtRatInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>

**Note**

: None

**8.812.2 Field Documentation**8.812.2.1 `swi_uint256_t` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::ParamPresenceMask`8.812.2.2 `imsa_SmsRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsRat`8.812.2.3 `imsa_SmsSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::SmsService`8.812.2.4 `imsa_UtRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtRat`8.812.2.5 `imsa_UtSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::UtService`8.812.2.6 `imsa_VoipRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipRat`8.812.2.7 `imsa_VoipSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VoipService`8.812.2.8 `imsa_VtRatInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtRat`8.812.2.9 `imsa_VtSvcStatusInfo` `unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t::VtService`**8.813 `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](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.813.1 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>xid</i>	<p>Transaction ID that was specified in the Get Best Available Position request.</p> <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>• Vertical uncertainty.</li> <li>• Units - Meters</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></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> <li>• Bit to check in ParamPresenceMask - <b>24</b></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> <li>• Bit to check in ParamPresenceMask - <b>25</b></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> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></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> <li>• Bit to check in ParamPresenceMask - <b>28</b></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> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> <li>• Horizontal speed.</li> <li>• Units - Meters/second</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> <li>• 3-D Speed uncertainty.</li> <li>• Units - Meters/second.</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>• Altitude With Respect to Sea Level.</li> <li>• Units - Meters</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> <li>• Vertical speed.</li> <li>• Units - Meters/second</li> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pSpeedVertical- Unc</i>	<ul style="list-style-type: none"> <li>• Vertical speed</li> <li>• Units: Meters/second</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></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> <li>• Bit to check in ParamPresenceMask - <b>37</b></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> <li>• Bit to check in ParamPresenceMask - <b>38</b></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> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>



<i>pTechnologyMask</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> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>-pPrecisionDilution</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_precisionDilution</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>-pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gpsTime</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>42</b></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> <li>• Bit to check in ParamPresenceMask - <b>43</b></li> </ul>
<i>-pSensorDataUsage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_sensorDataUsage</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>44</b></li> </ul>
<i>-pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_svUsedforFix</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>45</b></li> </ul>

## 8.813.2 Field Documentation

- 8.813.2.1 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtEllipsoid`
- 8.813.2.2 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.813.2.3 `swi_uint256_t unpack_loc_BestAvailPos_Ind_t::ParamPresenceMask`
- 8.813.2.4 `loc_gpsTime* unpack_loc_BestAvailPos_Ind_t::pGpsTime`
- 8.813.2.5 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeading`
- 8.813.2.6 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHeadingUnc`
- 8.813.2.7 `uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorCirConf`
- 8.813.2.8 `uint8_t* unpack_loc_BestAvailPos_Ind_t::pHorEllpConf`
- 8.813.2.9 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorReliability`
- 8.813.2.10 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncCircular`
- 8.813.2.11 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.813.2.12 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMajor`
- 8.813.2.13 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pHorUncEllipseSemiMinor`
- 8.813.2.14 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLatitude`
- 8.813.2.15 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pLongitude`
- 8.813.2.16 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pMagneticDeviation`
- 8.813.2.17 `loc_precisionDilution* unpack_loc_BestAvailPos_Ind_t::pPrecisionDilution`
- 8.813.2.18 `loc_sensorDataUsage* unpack_loc_BestAvailPos_Ind_t::pSensorDataUsage`
- 8.813.2.19 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedHorizontal`
- 8.813.2.20 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedUnc`
- 8.813.2.21 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVertical`
- 8.813.2.22 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pSpeedVerticalUnc`
- 8.813.2.23 `loc_svUsedforFix* unpack_loc_BestAvailPos_Ind_t::pSvUsedforFix`
- 8.813.2.24 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTechnologyMask`
- 8.813.2.25 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeSrc`
- 8.813.2.26 `uint64_t* unpack_loc_BestAvailPos_Ind_t::pTimestampUtc`
- 8.813.2.27 `uint32_t* unpack_loc_BestAvailPos_Ind_t::pTimeUnc`

8.813.2.28 uint8\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertConfidence

8.813.2.29 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertReliability

8.813.2.30 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pVertUnc

8.813.2.31 uint32\_t\* unpack\_loc\_BestAvailPos\_Ind\_t::pXid

8.813.2.32 uint32\_t unpack\_loc\_BestAvailPos\_Ind\_t::status

8.813.2.33 uint16\_t unpack\_loc\_BestAvailPos\_Ind\_t::Tlvresult

## 8.814 unpack\_loc\_CradleMountCallback\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [cradleMountConfigStatus](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.814.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
---------------------------------	--

### 8.814.2 Field Documentation

8.814.2.1 uint32\_t unpack\_loc\_CradleMountCallback\_Ind\_t::cradleMountConfigStatus

8.814.2.2 [swi\\_uint256\\_t](#) unpack\_loc\_CradleMountCallback\_Ind\_t::ParamPresenceMask

## 8.815 unpack\_loc\_Delete\_Assist\_Data\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.815.1 Detailed Description

This structure contains LOC delete assist data unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack result.</li><li>• Bit to check in ParamPresenceMask - 2</li></ul>
------------------	--

### 8.815.2 Field Documentation

8.815.2.1 [swi\\_uint256\\_t unpack\\_loc\\_Delete\\_Assist\\_Data\\_t::ParamPresenceMask](#)

8.815.2.2 [uint16\\_t unpack\\_loc\\_Delete\\_Assist\\_Data\\_t::Tlvresult](#)

## 8.816 unpack\_loc\_DeleteAssistData\_Ind\_t Struct Reference

### Data Fields

- [uint32\\_t status](#)
- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.816.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

## 8.816.2 Field Documentation

8.816.2.1 swi\_uint256\_t unpack\_loc\_DeleteAssistData\_Ind\_t::ParamPresenceMask

8.816.2.2 uint32\_t unpack\_loc\_DeleteAssistData\_Ind\_t::status

8.816.2.3 uint16\_t unpack\_loc\_DeleteAssistData\_Ind\_t::Tlvresult

## 8.817 unpack\_loc\_EngineState\_Ind\_t Struct Reference

## Data Fields

- uint32\_t [engineState](#)
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.817.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

## 8.817.2 Field Documentation

8.817.2.1 uint32\_t unpack\_loc\_EngineState\_Ind\_t::engineState

8.817.2.2 swi\_uint256\_t unpack\_loc\_EngineState\_Ind\_t::ParamPresenceMask

8.817.2.3 uint16\_t unpack\_loc\_EngineState\_Ind\_t::Tlvresult

## 8.818 unpack\_loc\_EventNMEA\_Ind\_t Struct Reference

## Data Fields

- char [NMEADData](#) [201]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.818.1 Detailed Description

Contain the parameters passed for SetLocEventMaskNMEACallback by the device.

## Parameters

<i>NMEADData</i>	<ul style="list-style-type: none"> <li>• NMEA string</li> <li>• Type - NULL terminated string</li> <li>• Maximum string length (including NULL terminator) - 201</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

## 8.818.2 Field Documentation

8.818.2.1 char unpack\_loc\_EventNMEA\_Ind\_t::NMEADData[201]

8.818.2.2 swi\_uint256\_t unpack\_loc\_EventNMEA\_Ind\_t::ParamPresenceMask

8.818.2.3 uint16\_t unpack\_loc\_EventNMEA\_Ind\_t::Tlvresult

## 8.819 unpack\_loc\_EventRegister\_t Struct Reference

### Data Fields

- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.819.1 Detailed Description

This structure contains Event Register unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"><li>• Unpack result.</li><li>• Bit to check in ParamPresenceMask - 2</li></ul>
------------------	--

### 8.819.2 Field Documentation

8.819.2.1 [swi\\_uint256\\_t unpack\\_loc\\_EventRegister\\_t::ParamPresenceMask](#)

8.819.2.2 [uint16\\_t unpack\\_loc\\_EventRegister\\_t::Tlvresult](#)

## 8.820 unpack\_loc\_EventTimeSyncCallback\_Ind\_t Struct Reference

### Data Fields

- [uint32\\_t timeSyncRefCounter](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.820.1 Detailed Description

This structure contains LOC Event Time Sync Reference COUNTER

#### Parameters

<i>timeSyncRef-Counter</i>	<ul style="list-style-type: none"><li>• Sent by the location engine when it needs to synchronize location engine and control point (sensor processor) times.</li><li>• Bit to check in ParamPresenceMask - 1</li></ul>
----------------------------	--

### 8.820.2 Field Documentation

8.820.2.1 [swi\\_uint256\\_t unpack\\_loc\\_EventTimeSyncCallback\\_Ind\\_t::ParamPresenceMask](#)

8.820.2.2 [uint32\\_t unpack\\_loc\\_EventTimeSyncCallback\\_Ind\\_t::timeSyncRefCounter](#)

## 8.821 unpack\_loc\_FixCriteria\_Ind\_t Struct Reference

## Data Fields

- [loc\\_FixCriteriaStatusTlv](#) [loc\\_fixCriteriaStatus](#)
- [loc\\_HorAccuracyLvlTlv](#) [loc\\_horAccuracy](#)
- [loc\\_IntermediateRptStateTlv](#) [loc\\_intermediateRptState](#)
- [loc\\_MinIntervalTlv](#) [loc\\_minInterval](#)
- [loc\\_AppProviderInfoTlv](#) [loc\\_appProviderInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.821.1 Detailed Description

Contain the parameters passed for SetLocGetFixCriteriaCallback by the device.

#### Parameters

<i>fixCriteriaStatus</i>	<ul style="list-style-type: none"> <li>• Status of the get fix criteria request</li> <li>• See <a href="#">loc_FixCriteriaStatusTlv</a> for more information.</li> </ul>
<i>horAccuracyLvl</i>	<ul style="list-style-type: none"> <li>• Horizontal accuracy level</li> <li>• See <a href="#">loc_HorAccuracyLvlTlv</a> for more information.</li> </ul>
<i>intermediateRpt-State</i>	<ul style="list-style-type: none"> <li>• Intermediate Report state (ON, OFF).</li> <li>• See <a href="#">loc_IntermediateRptStateTlv</a> for more information.</li> </ul>
<i>minInterval</i>	<ul style="list-style-type: none"> <li>• Minimum Interval Between fixes</li> <li>• See <a href="#">loc_MinIntervalTlv</a> for more information.</li> </ul>
<i>appProviderInfo</i>	<ul style="list-style-type: none"> <li>• ID of the Application that Sent the Position Request Application provider, name, and version.</li> <li>• See <a href="#">loc_AppProviderInfoTlv</a> for more information.</li> </ul>

#### Note

None

### 8.821.2 Field Documentation

8.821.2.1 [loc\\_AppProviderInfoTlv](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::loc\\_appProviderInfo](#)

8.821.2.2 [loc\\_FixCriteriaStatusTlv](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::loc\\_fixCriteriaStatus](#)

8.821.2.3 [loc\\_HorAccuracyLvlTlv](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::loc\\_horAccuracy](#)

8.821.2.4 [loc\\_IntermediateRptStateTlv](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::loc\\_intermediateRptState](#)

8.821.2.5 [loc\\_MinIntervalTlv](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::loc\\_minInterval](#)

8.821.2.6 [swi\\_uint256\\_t](#) [unpack\\_loc\\_FixCriteria\\_Ind\\_t::ParamPresenceMask](#)



## 8.822 unpack\_loc\_GetOpMode\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [Status](#)
- uint32\_t \* [pMode](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.822.1 Detailed Description

Contain the parameters passed for SetLocGetOpModeCallback by the device.

#### Parameters

<i>Status</i>	<ul style="list-style-type: none"> <li>• Status of the Get Server 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> <li>– 0xffffffff - Invalid data.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pMode</i>	<ul style="list-style-type: none"> <li>• Current operation mode.</li> <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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.822.2 Field Documentation

8.822.2.1 `swi_uint256_t unpack_loc_GetOpMode_Ind_t::ParamPresenceMask`

8.822.2.2 `uint32_t* unpack_loc_GetOpMode_Ind_t::pMode`

8.822.2.3 `uint32_t unpack_loc_GetOpMode_Ind_t::Status`

## 8.823 `unpack_loc_GetServer_Ind_t` Struct Reference

### Data Fields

- `uint32_t` [serverStatus](#)
- `uint32_t` [serverType](#)
- `loc_IPv4Info * pIPv4AddrInfo`
- `loc_IPv6Info * pIPv6AddrInfo`
- `loc_urlAddr * pURL`
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.823.1 Detailed Description

Contain the parameters passed for `SetLocGetServerCallback` by the device.

#### Parameters

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

<i>serverType</i>	<ul style="list-style-type: none"> <li>Type of server.</li> <li>Valid values <ul style="list-style-type: none"> <li>eQMI_LOC_SERVER_TYPE_CDMA_PDE (1) - Server type is CDMA PDE</li> <li>eQMI_LOC_SERVER_TYPE_CDMA_MPC (2) - Server type is CDMA MPC</li> <li>eQMI_LOC_SERVER_TYPE_UMTS_SLP (3) - Server type is UMTS SLP</li> <li>eQMI_LOC_SERVER_TYPE_CUSTOM_PDE (4) - Server type is custom PDE</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>2</b></li> </ul>
<i>pIPv4AddrInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">loc_IPv4Info</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pIPv6AddrInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">loc_IPv6Info</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pURL</i>	<ul style="list-style-type: none"> <li>See <a href="#">loc_urlAddr</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

## 8.823.2 Field Documentation

8.823.2.1 `swi_uint256_t unpack_loc_GetServer_Ind_t::ParamPresenceMask`

8.823.2.2 `loc_IPv4Info* unpack_loc_GetServer_Ind_t::pIPv4AddrInfo`

8.823.2.3 `loc_IPv6Info* unpack_loc_GetServer_Ind_t::pIPv6AddrInfo`

8.823.2.4 `loc_urlAddr* unpack_loc_GetServer_Ind_t::pURL`

8.823.2.5 `uint32_t unpack_loc_GetServer_Ind_t::serverStatus`

8.823.2.6 `uint32_t unpack_loc_GetServer_Ind_t::serverType`

8.823.2.7 `uint16_t unpack_loc_GetServer_Ind_t::Tlvresult`

## 8.824 unpack\_loc\_GnssSvInfo\_Ind\_t Struct Reference

### Data Fields

- `uint8_t altitudeAssumed`
- `loc_satelliteInfo * pSatelliteInfo`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.824.1 Detailed Description

Contain the parameters passed for SetLocGnssSvInfoCallback by the device.

## Parameters

<i>altitudeAssumed</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pSatelliteInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">loc_satelliteInfo</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>

## 8.824.2 Field Documentation

8.824.2.1 `uint8_t unpack_loc_GnssSvInfo_Ind_t::altitudeAssumed`8.824.2.2 `swi_uint256_t unpack_loc_GnssSvInfo_Ind_t::ParamPresenceMask`8.824.2.3 `loc_satelliteInfo* unpack_loc_GnssSvInfo_Ind_t::pSatelliteInfo`8.824.2.4 `uint16_t unpack_loc_GnssSvInfo_Ind_t::Tlvresult`8.825 `unpack_loc_InjectPositionCallback_Ind_t` Struct Reference

## Data Fields

- `uint32_t` [status](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.825.1 Detailed Description

Contain the parameters passed for LOC Inject Position indication by the device.

## Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Position Injection Status</li> <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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
---------------	--

## Note

None

## 8.825.2 Field Documentation

8.825.2.1 swi\_uint256\_t unpack\_loc\_InjectPositionCallback\_Ind\_t::ParamPresenceMask

8.825.2.2 uint32\_t unpack\_loc\_InjectPositionCallback\_Ind\_t::status

## 8.826 unpack\_loc\_InjectSensorDataCallback\_Ind\_t Struct Reference

## Data Fields

- uint32\_t [injectSensorDataStatus](#)
- uint32\_t \* [pOpaqueIdentifier](#)
- uint8\_t \* [pAccelSamplesAccepted](#)
- uint8\_t \* [pGyroSamplesAccepted](#)
- uint8\_t \* [pAccelTempSamplesAccepted](#)
- uint8\_t \* [pGyroTempSamplesAccepted](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.826.1 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pOpaque-Identifier[OPTIONAL]</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> <li>• NULL when this TLV not present</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pAccelSamples-Accepted[OPTIONAL]</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> <li>• NULL when this TLV not present</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pGyroSamples-Accepted[OPTIONAL]</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> <li>• NULL when this TLV not present</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pAccelTemp-Samples-Accepted[OPTIONAL]</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> <li>• NULL when this TLV not present</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pGyroTemp-Samples-Accepted[OPTIONAL]</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> <li>• NULL when this TLV not present</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>

## 8.826.2 Field Documentation

8.826.2.1 uint32\_t unpack\_loc\_InjectSensorDataCallback\_Ind\_t::injectSensorDataStatus

8.826.2.2 uint8\_t\* unpack\_loc\_InjectSensorDataCallback\_Ind\_t::pAccelSamplesAccepted

8.826.2.3 uint8\_t\* unpack\_loc\_InjectSensorDataCallback\_Ind\_t::pAccelTempSamplesAccepted

8.826.2.4 swi\_uint256\_t unpack\_loc\_InjectSensorDataCallback\_Ind\_t::ParamPresenceMask

8.826.2.5 uint8\_t\* unpack\_loc\_InjectSensorDataCallback\_Ind\_t::pGyroSamplesAccepted

8.826.2.6 uint8\_t\* unpack\_loc\_InjectSensorDataCallback\_Ind\_t::pGyroTempSamplesAccepted

8.826.2.7 uint32\_t\* unpack\_loc\_InjectSensorDataCallback\_Ind\_t::pOpaqueldentifier

## 8.827 unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [injectTimeSyncStatus](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.827.1 Detailed Description

This structure contains LOC Inject Time Sync Data Status

#### Parameters

<i>injectTimeSync-Status</i>	<ul style="list-style-type: none"> <li>• Values             <ul style="list-style-type: none"> <li>– 0 - Request was completed successfully</li> <li>– 1 - Request failed because of a general failure.</li> <li>– 2 - Request failed because it is not supported.</li> <li>– 3 - Request failed because it contained invalid parameters</li> <li>– 4 - Request failed because the engine is busy</li> <li>– 5 - Request failed because the phone is offline</li> <li>– 6 - Request failed because it timed out</li> <li>– 7 - Request failed because an undefined configuration was requested</li> <li>– 8 - engine could not allocate sufficient memory</li> <li>– 9 - Request failed because the maximum number of Geofences are already programmed</li> <li>– 10 -Location service failed because of an XTRA version-based file format check failure</li> </ul> </li> </ul>
------------------------------	---

- \* - Bit to check in ParamPresenceMask - 1

### 8.827.2 Field Documentation

8.827.2.1 uint32\_t unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t::injectTimeSyncStatus

8.827.2.2 swi\_uint256\_t unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t::ParamPresenceMask

## 8.828 unpack\_loc\_InjectUTCTimeCallback\_Ind\_t Struct Reference

### Data Fields

- uint32\_t [status](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.828.1 Detailed Description

Contain the parameters passed for Inject UTC time indication 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>– 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
---------------	---

#### Note

None

### 8.828.2 Field Documentation

8.828.2.1 `swi_uint256_t unpack_loc_InjectUTCtimeCallback_Ind_t::ParamPresenceMask`

8.828.2.2 `uint32_t unpack_loc_InjectUTCtimeCallback_Ind_t::status`

## 8.829 `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
- swi\_uint256\_t ParamPresenceMask

### 8.829.1 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>2</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pHorUncCircular</i>	<ul style="list-style-type: none"> <li>• Horizontal position uncertainty.</li> <li>• Units - Meters</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></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> <li>• Bit to check in ParamPresenceMask - <b>22</b></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>

- Bit to check in ParamPresenceMask - **23**

## Parameters

<i>pSpeed-Horizontal</i>	<ul style="list-style-type: none"> <li>Horizontal speed.</li> <li>Units - Meters/second</li> <li>Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pSpeedUnc</i>	<ul style="list-style-type: none"> <li>3-D Speed uncertainty.</li> <li>Units - Meters/second.</li> <li>Bit to check in ParamPresenceMask - <b>25</b></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> <li>Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pAltitudeWrt-MeanSeaLevel</i>	<ul style="list-style-type: none"> <li>Altitude With Respect to Sea Level.</li> <li>Units - Meters</li> <li>Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pVertUnc</i>	<ul style="list-style-type: none"> <li>Vertical uncertainty.</li> <li>Units - Meters</li> <li>Bit to check in ParamPresenceMask - <b>28</b></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> <li>Bit to check in ParamPresenceMask - <b>29</b></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> <li>Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pSpeedVertical</i>	<ul style="list-style-type: none"> <li>Vertical speed.</li> <li>Units - Meters/second</li> <li>Bit to check in ParamPresenceMask - <b>31</b></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> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pPrecision-Dilution</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_precisionDilution</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>pTimestampUtc</i>	<ul style="list-style-type: none"> <li>• UTC timestamp</li> <li>• Units - Milliseconds since Jan. 1, 1970</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></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> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pGpsTime</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gpsTime</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>

<i>pTimeUnc</i>	<ul style="list-style-type: none"> <li>• Time uncertainty.</li> <li>• Units - Milliseconds</li> <li>• Bit to check in ParamPresenceMask - <b>40</b></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>

- Bit to check in ParamPresenceMask - **41**

#### Parameters

<i>pSensorData-Usage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_sensorDataUsage</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>42</b></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> <li>• Bit to check in ParamPresenceMask - <b>43</b></li> </ul>
<i>pSvUsedforFix</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_svUsedforFix</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>44</b></li> </ul>
<i>pAltitude-Assumed</i>	<ul style="list-style-type: none"> <li>• Indicates whether altitude is assumed or calculated.</li> <li>• Value <ul style="list-style-type: none"> <li>– 0x00 - Altitude is calculated</li> <li>– 0x01 - Altitude is assumed</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>45</b></li> </ul>

## 8.829.2 Field Documentation

- 8.829.2.1 `uint8_t* unpack_loc_PositionRpt_Ind_t::pAltitudeAssumed`
- 8.829.2.2 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtEllipsoid`
- 8.829.2.3 `uint32_t* unpack_loc_PositionRpt_Ind_t::pAltitudeWrtMeanSeaLevel`
- 8.829.2.4 `swi_uint256_t unpack_loc_PositionRpt_Ind_t::ParamPresenceMask`
- 8.829.2.5 `uint32_t* unpack_loc_PositionRpt_Ind_t::pFixId`
- 8.829.2.6 `loc_gpsTime* unpack_loc_PositionRpt_Ind_t::pGpsTime`
- 8.829.2.7 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeading`
- 8.829.2.8 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHeadingUnc`
- 8.829.2.9 `uint8_t* unpack_loc_PositionRpt_Ind_t::pHorConfidence`
- 8.829.2.10 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorReliability`
- 8.829.2.11 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncCircular`
- 8.829.2.12 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseOrientAzimuth`
- 8.829.2.13 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMajor`
- 8.829.2.14 `uint32_t* unpack_loc_PositionRpt_Ind_t::pHorUncEllipseSemiMinor`
- 8.829.2.15 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLatitude`
- 8.829.2.16 `uint8_t* unpack_loc_PositionRpt_Ind_t::pLeapSeconds`
- 8.829.2.17 `uint64_t* unpack_loc_PositionRpt_Ind_t::pLongitude`
- 8.829.2.18 `uint32_t* unpack_loc_PositionRpt_Ind_t::pMagneticDeviation`
- 8.829.2.19 `loc_precisionDilution* unpack_loc_PositionRpt_Ind_t::pPrecisionDilution`
- 8.829.2.20 `loc_sensorDataUsage* unpack_loc_PositionRpt_Ind_t::pSensorDataUsage`
- 8.829.2.21 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedHorizontal`
- 8.829.2.22 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedUnc`
- 8.829.2.23 `uint32_t* unpack_loc_PositionRpt_Ind_t::pSpeedVertical`
- 8.829.2.24 `loc_svUsedforFix* unpack_loc_PositionRpt_Ind_t::pSvUsedforFix`
- 8.829.2.25 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTechnologyMask`
- 8.829.2.26 `uint32_t* unpack_loc_PositionRpt_Ind_t::pTimeSrc`
- 8.829.2.27 `uint64_t* unpack_loc_PositionRpt_Ind_t::pTimestampUtc`

- 8.829.2.28 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pTimeUnc
- 8.829.2.29 uint8\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertConfidence
- 8.829.2.30 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertReliability
- 8.829.2.31 uint32\_t\* unpack\_loc\_PositionRpt\_Ind\_t::pVertUnc
- 8.829.2.32 uint8\_t unpack\_loc\_PositionRpt\_Ind\_t::sessionId
- 8.829.2.33 uint32\_t unpack\_loc\_PositionRpt\_Ind\_t::sessionStatus
- 8.829.2.34 uint16\_t unpack\_loc\_PositionRpt\_Ind\_t::Tlvresult

## 8.830 unpack\_loc\_SensorStreamingCallback\_Ind\_t Struct Reference

### Data Fields

- [loc\\_accelAcceptReady](#) \* [pAccelAcceptReady](#)
- [loc\\_gyroAcceptReady](#) \* [pGyroAcceptReady](#)
- [loc\\_accelTempAcceptReady](#) \* [pAccelTempAcceptReady](#)
- [loc\\_gyroTempAcceptReady](#) \* [pGyroTempAcceptReady](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.830.1 Detailed Description

This structure contains LOC Event Sensor Streaming Ready Status

#### Parameters

<i>-pAccelAcceptReady[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_accelAcceptReady</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>-pGyroAcceptReady[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gyroAcceptReady</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>-pAccelTempAcceptReady[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_accelTempAcceptReady</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>-pGyroTempAcceptReady[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">loc_gyroTempAcceptReady</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

### 8.830.2 Field Documentation

- 8.830.2.1 [loc\\_accelAcceptReady](#)\* [unpack\\_loc\\_SensorStreamingCallback\\_Ind\\_t::pAccelAcceptReady](#)
- 8.830.2.2 [loc\\_accelTempAcceptReady](#)\* [unpack\\_loc\\_SensorStreamingCallback\\_Ind\\_t::pAccelTempAcceptReady](#)

8.830.2.3 `swi_uint256_t unpack_loc_SensorStreamingCallback_Ind_t::ParamPresenceMask`

8.830.2.4 `loc_gyroAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroAcceptReady`

8.830.2.5 `loc_gyroTempAcceptReady* unpack_loc_SensorStreamingCallback_Ind_t::pGyroTempAcceptReady`

## 8.831 `unpack_loc_SetExtPowerConfig_Ind_t` Struct Reference

### Data Fields

- `uint32_t status`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.831.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.831.2 Field Documentation

8.831.2.1 `swi_uint256_t unpack_loc_SetExtPowerConfig_Ind_t::ParamPresenceMask`

8.831.2.2 `uint32_t unpack_loc_SetExtPowerConfig_Ind_t::status`

8.831.2.3 `uint16_t unpack_loc_SetExtPowerConfig_Ind_t::Tlvresult`

## 8.832 `unpack_loc_SetExtPowerState_t` Struct Reference

### Data Fields

- `uint16_t Tlvresult`



- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.832.1 Detailed Description

This structure contains Set Ext Power State unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
------------------	---

### 8.832.2 Field Documentation

8.832.2.1 [swi\\_uint256\\_t unpack\\_loc\\_SetExtPowerState\\_t::ParamPresenceMask](#)

8.832.2.2 [uint16\\_t unpack\\_loc\\_SetExtPowerState\\_t::Tlvresult](#)

## 8.833 unpack\_loc\_SetOperationMode\_Ind\_t Struct Reference

### Data Fields

- [uint32\\_t status](#)
- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.833.1 Detailed Description

This structure contains LOC Set External Power Configure status field.

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• Status of the Set Operation Mode request.</li> <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 - Request failed because the engine could not allocate sufficient memory for the request</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>

### 8.833.2 Field Documentation

8.833.2.1 `swi_uint256_t unpack_loc_SetOperationMode_Ind_t::ParamPresenceMask`

8.833.2.2 `uint32_t unpack_loc_SetOperationMode_Ind_t::status`

8.833.2.3 `uint16_t unpack_loc_SetOperationMode_Ind_t::Tlvresult`

## 8.834 `unpack_loc_SetOperationMode_t` Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.834.1 Detailed Description

This structure contains Set Operation Mode unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
------------------	---

### 8.834.2 Field Documentation

8.834.2.1 `swi_uint256_t unpack_loc_SetOperationMode_t::ParamPresenceMask`

8.834.2.2 `uint16_t unpack_loc_SetOperationMode_t::Tlvresult`

## 8.835 `unpack_loc_SetServer_Ind_t` Struct Reference

### Data Fields

- `uint32_t serverStatus`
- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.835.1 Detailed Description

Contain the parameters passed for SetLocSetServerCallback by the device.

## Parameters

<i>serverStatus</i>	<ul style="list-style-type: none"> <li>• Status of the Get Server 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> <li>– 0xffffffff - Invalid data.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.835.2 Field Documentation

8.835.2.1 swi\_uint256\_t unpack\_loc\_SetServer\_Ind\_t::ParamPresenceMask

8.835.2.2 uint32\_t unpack\_loc\_SetServer\_Ind\_t::serverStatus

8.835.2.3 uint16\_t unpack\_loc\_SetServer\_Ind\_t::Tlvresult

## 8.836 unpack\_loc\_SLQSLOCGetBestAvailPos\_t Struct Reference

## Data Fields

- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.836.1 Detailed Description

This structure contains Set Operation Mode unpack

## Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
------------------	---

## 8.836.2 Field Documentation

8.836.2.1 `swi_uint256_t unpack_loc_SLQSLOCGetBestAvailPos_t::ParamPresenceMask`

8.836.2.2 `uint16_t unpack_loc_SLQSLOCGetBestAvailPos_t::Tlvresult`

## 8.837 unpack\_loc\_SLQSLOCGetOpMode\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.837.1 Detailed Description

This structure contains Start LOC unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.837.2 Field Documentation

8.837.2.1 `swi_uint256_t unpack_loc_SLQSLOCGetOpMode_t::ParamPresenceMask`

8.837.2.2 `uint16_t unpack_loc_SLQSLOCGetOpMode_t::Tlvresult`

## 8.838 unpack\_loc\_Start\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.838.1 Detailed Description

This structure contains Start LOC unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
------------------	---

### 8.838.2 Field Documentation

8.838.2.1 `swi_uint256_t unpack_loc_Start_t::ParamPresenceMask`

8.838.2.2 `uint16_t unpack_loc_Start_t::Tlvresult`

## 8.839 unpack\_loc\_Stop\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.839.1 Detailed Description

This structure contains Stop LOC unpack

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack result.</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
------------------	---

### 8.839.2 Field Documentation

8.839.2.1 `swi_uint256_t unpack_loc_Stop_t::ParamPresenceMask`

8.839.2.2 `uint16_t unpack_loc_Stop_t::Tlvresult`

## 8.840 unpack\_nas\_GetACCOLC\_t Struct Reference

### Data Fields

- `uint8_t * pAccolc`
- `swi_uint256_t ParamPresenceMask`

### 8.840.1 Detailed Description

Retrieves information about the access overload class unpack structure

#### Parameters

<i>pAccolc</i>	<ul style="list-style-type: none"> <li>• An 8-bit integer representation of the ACCOLC. <ul style="list-style-type: none"> <li>– Range: 0 to 15 (0x00 to 0x0F).</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.840.2 Field Documentation

8.840.2.1 `uint8_t*` `unpack_nas_GetACCOLC_t::pAccolc`

8.840.2.2 `swi_uint256_t` `unpack_nas_GetACCOLC_t::ParamPresenceMask`

## 8.841 `unpack_nas_GetANAAAAuthenticationStatus_t` Struct Reference

### Data Fields

- `uint32_t` \* `pAuthStatus`
- `swi_uint256_t` `ParamPresenceMask`

### 8.841.1 Detailed Description

AN-AAA authentication status of the device

#### Parameters

<i>AN-AAA</i>	Authentication Status. <ul style="list-style-type: none"> <li>• Status of the last AN-AAA authentication request, if any, for the current 1xEV-DO session.               <ul style="list-style-type: none"> <li>– 0 - AAA_STATUS_FAILED - Authentication failed</li> <li>– 1 - AAA_STATUS_SUCCESS - Authentication success</li> <li>– 2 - AAA_STATUS_NO_REQUEST - No authentication requested</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.841.2 Field Documentation

8.841.2.1 `swi_uint256_t` `unpack_nas_GetANAAAAuthenticationStatus_t::ParamPresenceMask`

8.841.2.2 `uint32_t*` `unpack_nas_GetANAAAAuthenticationStatus_t::pAuthStatus`

## 8.842 `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`
- `swi_uint256_t` `ParamPresenceMask`

## 8.842.1 Detailed Description

Gets the current CDMA network parameters

## Parameters

<i>SCI</i>	<ul style="list-style-type: none"> <li>• Slot cycle index</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>SCM</i>	<ul style="list-style-type: none"> <li>• Station class mark</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>RegHomeSID</i>	<ul style="list-style-type: none"> <li>• Register on home SID <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>RegForeignSID</i>	<ul style="list-style-type: none"> <li>• Register on foreign SID <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>RegForeignNID</i>	<ul style="list-style-type: none"> <li>• Register on foreign NID <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ForceRev0</i>	<ul style="list-style-type: none"> <li>• Force CDMA 1x-EV-DO Rev. 0 mode <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>CustomSCP</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>

<i>Protocol</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 Port</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>Broadcast</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>Application</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>Roaming</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.842.2 Field Documentation

8.842.2.1 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Application

8.842.2.2 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Broadcast

8.842.2.3 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::CustomSCP

8.842.2.4 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::ForceRev0

8.842.2.5 swi\_uint256\_t unpack\_nas\_GetCDMANetworkParameters\_t::ParamPresenceMask

8.842.2.6 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Protocol

8.842.2.7 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignNID



8.842.2.8 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegForeignSID

8.842.2.9 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::RegHomeSID

8.842.2.10 uint32\_t unpack\_nas\_GetCDMANetworkParameters\_t::Roaming

8.842.2.11 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCI

8.842.2.12 uint8\_t unpack\_nas\_GetCDMANetworkParameters\_t::SCM

## 8.843 unpack\_nas\_GetHomeNetwork3GPP2\_t Struct Reference

### Data Fields

- uint16\_t \* pMCC
- uint16\_t \* pMNC
- uint8\_t nameSize
- uint8\_t \* pName
- uint16\_t \* pSID
- uint16\_t \* pNID
- uint16\_t \* pNw2MCC
- uint16\_t \* pNw2MNC
- uint8\_t \* pNw2DescDisp
- uint8\_t \* pNw2DescEnc
- uint8\_t \* pNw2DescLen
- uint8\_t \* pNw2Name
- swi\_uint256\_t ParamPresenceMask

### 8.843.1 Detailed Description

Structure for HomeNetwork3GPP2 unpack.

#### Parameters

<i>pMCC</i>	<ul style="list-style-type: none"> <li>• Mobile country code (UMTS only).</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pMNC</i>	<ul style="list-style-type: none"> <li>• Mobile network code (UMTS only).</li> <li>• Bit to check in ParamPresenceMask - 1</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 (UMTS only).</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pName</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>

<i>pSID</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pNID</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pNw2MCC</i>	<ul style="list-style-type: none"> <li>• Mobile country code (3GPP2 only).</li> <li>• Range : 0 to 999</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNw2MNC</i>	<ul style="list-style-type: none"> <li>• Mobile network code (3GPP2 only).</li> <li>• Range : 0 to 999</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNw2DescDisp</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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNw2DescEnc</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 (<a href="#">liteqmi_helper_decode7bitAsciiEncString</a>)</li> <li>– 0x04 - Unicode</li> <li>– 0x09 - GSM 7-bit default</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNw2DescLen</i>	<ul style="list-style-type: none"> <li>• Network Description Length (3GPP2 only).</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNw2Name</i>	<ul style="list-style-type: none"> <li>• Network Name (3GPP2 only).</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.843.2 Field Documentation

8.843.2.1 `uint8_t unpack_nas_GetHomeNetwork3GPP2_t::nameSize`

8.843.2.2 `swi_uint256_t unpack_nas_GetHomeNetwork3GPP2_t::ParamPresenceMask`

8.843.2.3 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMCC`

8.843.2.4 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pMNC`

8.843.2.5 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pName`

8.843.2.6 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNID`

8.843.2.7 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescDisp`

8.843.2.8 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescEnc`

8.843.2.9 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2DescLen`

8.843.2.10 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MCC`

8.843.2.11 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2MNC`

8.843.2.12 `uint8_t* unpack_nas_GetHomeNetwork3GPP2_t::pNw2Name`

8.843.2.13 `uint16_t* unpack_nas_GetHomeNetwork3GPP2_t::pSID`

## 8.844 unpack\_nas\_GetHomeNetwork\_t Struct Reference

### Data Fields

- `uint16_t mcc`
- `uint16_t mnc`
- `char name` [255]
- `uint16_t sid`
- `uint16_t nid`
- `swi_uint256_t ParamPresenceMask`

### 8.844.1 Detailed Description

This structure contains unpack get get home network parameters.

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"> <li>• Mobile country code (UMTS only).</li> <li>• A 16-bit integer representation of MCC. Range: 0 to 999.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>• Mobile network code (UMTS only).</li> <li>• A 16-bit integer representation of MNC. Range: 0 to 999.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>

<i>name</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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>sid</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>nid</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.844.2 Field Documentation

8.844.2.1 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mcc

8.844.2.2 uint16\_t unpack\_nas\_GetHomeNetwork\_t::mnc

8.844.2.3 char unpack\_nas\_GetHomeNetwork\_t::name[255]

8.844.2.4 uint16\_t unpack\_nas\_GetHomeNetwork\_t::nid

8.844.2.5 swi\_uint256\_t unpack\_nas\_GetHomeNetwork\_t::ParamPresenceMask

8.844.2.6 uint16\_t unpack\_nas\_GetHomeNetwork\_t::sid

## 8.845 unpack\_nas\_GetNetworkPreference\_t Struct Reference

### Data Fields

- uint32\_t [ActiveTechPref](#)
- uint32\_t [Duration](#)
- uint32\_t [PersistentTechPref](#)
- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.845.1 Detailed Description

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 [unpack\\_nas\\_SLQSGetSysSelectionPref\(\)](#) for new firmware versions and new modules

## 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.845.2 Field Documentation

8.845.2.1 uint32\_t unpack\_nas\_GetNetworkPreference\_t::ActiveTechPref

8.845.2.2 uint32\_t unpack\_nas\_GetNetworkPreference\_t::Duration

8.845.2.3 swi\_uint256\_t unpack\_nas\_GetNetworkPreference\_t::ParamPresenceMask

8.845.2.4 uint32\_t unpack\_nas\_GetNetworkPreference\_t::PersistentTechPref

8.845.2.5 uint16\_t unpack\_nas\_GetNetworkPreference\_t::Tlvresult

## 8.846 unpack\_nas\_GetRFInfo\_t Struct Reference

## Data Fields

- `uint8_t instancesSize`
- `RFBandInfoElements RFBandInfoElements [255]`
- `swi_uint256_t ParamPresenceMask`

### 8.846.1 Detailed Description

Structur to store the RFInfoList

#### Parameters

<i>instanceSize</i>	<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>RFBandInfo-Elements</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.846.2 Field Documentation

8.846.2.1 `uint8_t unpack_nas_GetRFInfo_t::instancesSize`

8.846.2.2 `swi_uint256_t unpack_nas_GetRFInfo_t::ParamPresenceMask`

8.846.2.3 `RFBandInfoElements unpack_nas_GetRFInfo_t::RFBandInfoElements[255]`

## 8.847 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]`
- `swi_uint256_t ParamPresenceMask`

### 8.847.1 Detailed Description

This structure contains unpack get get serving network parameters.

#### Parameters

<i>Registration-State</i>	<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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>CSDomain</i>	<ul style="list-style-type: none"> <li>Circuit switch domain status: <ul style="list-style-type: none"> <li>0 - Unknown/Not Applicable</li> <li>1 - Attached</li> <li>2 - Detached</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>PSDomain</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>RAN</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>RadiolfacesSize</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>Radiolfaces</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>Roaming</i>	<ul style="list-style-type: none"> <li>Roaming indicator</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>MCC</i>	<ul style="list-style-type: none"> <li>Mobile country code</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>MNC</i>	<ul style="list-style-type: none"> <li>• Mobile network code</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>Name</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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>DataCapsLen[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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>DataCaps[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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.847.2 Field Documentation

8.847.2.1 uint32\_t unpack\_nas\_GetServingNetwork\_t::CSDomain

8.847.2.2 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCaps[255]

8.847.2.3 uint8\_t unpack\_nas\_GetServingNetwork\_t::DataCapsLen

8.847.2.4 uint16\_t unpack\_nas\_GetServingNetwork\_t::MCC

8.847.2.5 uint16\_t unpack\_nas\_GetServingNetwork\_t::MNC



- 8.847.2.6 uint8\_t unpack\_nas\_GetServingNetwork\_t::Name[255]
- 8.847.2.7 uint8\_t unpack\_nas\_GetServingNetwork\_t::nameSize
- 8.847.2.8 swi\_uint256\_t unpack\_nas\_GetServingNetwork\_t::ParamPresenceMask
- 8.847.2.9 uint32\_t unpack\_nas\_GetServingNetwork\_t::PSDomain
- 8.847.2.10 uint8\_t unpack\_nas\_GetServingNetwork\_t::Radiofaces[255]
- 8.847.2.11 uint8\_t unpack\_nas\_GetServingNetwork\_t::RadiofacesSize
- 8.847.2.12 uint32\_t unpack\_nas\_GetServingNetwork\_t::RAN
- 8.847.2.13 uint32\_t unpack\_nas\_GetServingNetwork\_t::RegistrationState
- 8.847.2.14 uint32\_t unpack\_nas\_GetServingNetwork\_t::Roaming

## 8.848 unpack\_nas\_GetServingNetworkCapabilities\_t Struct Reference

### Data Fields

- uint8\_t [DataCapsLen](#)
- uint8\_t [DataCaps](#) [255]
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.848.1 Detailed Description

This structure contains unpack get serving network capabilities parameters.

#### Parameters

<i>DataCapsLen</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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>DataCaps[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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.848.2 Field Documentation

8.848.2.1 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCaps[255]`

8.848.2.2 `uint8_t unpack_nas_GetServingNetworkCapabilities_t::DataCapsLen`

8.848.2.3 `swi_uint256_t unpack_nas_GetServingNetworkCapabilities_t::ParamPresenceMask`

## 8.849 `unpack_nas_GetSignalStrengths_t` Struct Reference

### Data Fields

- `uint32_t len`
- signed char `rsi` [8]
- `uint32_t radio` [8]
- `swi_uint256_t ParamPresenceMask`

### 8.849.1 Detailed Description

This structure contains unpack get signal strengths parameters.

#### Parameters

<i>len</i>	<ul style="list-style-type: none"> <li>number of rssi &amp; radio items following</li> </ul>
<i>rsi</i>	<ul style="list-style-type: none"> <li>signal strength array</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>radio</i>	<ul style="list-style-type: none"> <li>radio interface array</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.849.2 Field Documentation

8.849.2.1 `uint32_t unpack_nas_GetSignalStrengths_t::len`

8.849.2.2 `swi_uint256_t unpack_nas_GetSignalStrengths_t::ParamPresenceMask`

8.849.2.3 `uint32_t unpack_nas_GetSignalStrengths_t::radio[8]`

8.849.2.4 `signed char unpack_nas_GetSignalStrengths_t::rsi[8]`

## 8.850 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](#)
- [nas\\_QmisNasSlqsNasPCIInfo](#) \* [pPCIInfo](#)
- [nas\\_lteOpModeTlv](#) \* [pLteOpModeTlv](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.850.1 Detailed Description

Contain the network scan information.

#### Parameters

<i>p3GppNetwork-InstanceSize</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>p3GppNetwork-InfoInstances</i>	<ul style="list-style-type: none"> <li>• Network info instance array <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmiNas3GppNetworkInfo</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pRATInstance-Size</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pRATInstance</i>	<ul style="list-style-type: none"> <li>• RAT info instance array <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmiNas3GppNetworkRAT</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pPCSInstance-Size</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pPCSInstance</i>	<ul style="list-style-type: none"> <li>• PCS Digit info instance array <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmisNasPcsDigit</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>

<i>pScanResult</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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pPCIInfo</i>	<ul style="list-style-type: none"> <li>• PCI Information <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmisNasSlqsNasPCIInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pLteOpModeTlv[OUT]</i>	<ul style="list-style-type: none"> <li>• LTE Operational Mode.</li> <li>• See <a href="#">nas_LteOpModeTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.850.2 Field Documentation

8.850.2.1 `nas_QmiNas3GppNetworkInfo*` `unpack_nas_PerformNetworkScan_t::p3GppNetworkInfoInstances`

8.850.2.2 `uint8_t*` `unpack_nas_PerformNetworkScan_t::p3GppNetworkInstanceSize`

8.850.2.3 `swi_uint256_t` `unpack_nas_PerformNetworkScan_t::ParamPresenceMask`

8.850.2.4 `nas_LteOpModeTlv*` `unpack_nas_PerformNetworkScan_t::pLteOpModeTlv`

8.850.2.5 `nas_QmisNasSlqsNasPCIInfo*` `unpack_nas_PerformNetworkScan_t::pPCIInfo`

8.850.2.6 `nas_QmisNasPcsDigit*` `unpack_nas_PerformNetworkScan_t::pPCSInstance`

8.850.2.7 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pPCSInstanceSize`

8.850.2.8 `nas_QmiNas3GppNetworkRAT*` `unpack_nas_PerformNetworkScan_t::pRATInstance`

8.850.2.9 `uint8_t*` `unpack_nas_PerformNetworkScan_t::pRATInstanceSize`

8.850.2.10 `uint32_t*` `unpack_nas_PerformNetworkScan_t::pScanResult`

## 8.851 `unpack_nas_SetDataCapabilitiesCallback_ind_t` Struct Reference

### Data Fields

- `uint8_t` `dataCapsSize`
- `uint8_t` `dataCaps` [255]
- `swi_uint256_t` `ParamPresenceMask`

### 8.851.1 Detailed Description

This structure contains unpack set data capabilities callback indication parameters.

#### Parameters

<i>dataCapsSize</i>	<ul style="list-style-type: none"> <li>• Number of Data Capabilities</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>dataCaps</i>	<ul style="list-style-type: none"> <li>• Data Capabilities</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.851.2 Field Documentation

8.851.2.1 `uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCaps[255]`

8.851.2.2 `uint8_t unpack_nas_SetDataCapabilitiesCallback_ind_t::dataCapsSize`

8.851.2.3 `swi_uint256_t unpack_nas_SetDataCapabilitiesCallback_ind_t::ParamPresenceMask`

## 8.852 unpack\_nas\_SetEventReportInd\_t Struct Reference

### Data Fields

- [nas\\_SignalStrengthTlv SSTlv](#)
- [nas\\_RFInfoTlv RFTlv](#)
- [nas\\_RejectReasonTlv RRTlv](#)
- [nas\\_SLQSSignalStrengthsTlv SLQSSSTlv](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.852.1 Detailed Description

This structure contains unpack set event report indication parameters.

#### Parameters

<i>SSTlv</i>	<ul style="list-style-type: none"> <li>• signal strength tlv</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>RFTlv</i>	<ul style="list-style-type: none"> <li>• RF tlv</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>RRTlv</i>	<ul style="list-style-type: none"> <li>• RR tlv</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>SLQSSSTlv</i>	<ul style="list-style-type: none"> <li>• signal strength complete info tlv</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.852.2 Field Documentation

8.852.2.1 `swi_uint256_t` `unpack_nas_SetEventReportInd_t::ParamPresenceMask`

8.852.2.2 `nas_RFInfoTlv` `unpack_nas_SetEventReportInd_t::RFTlv`

8.852.2.3 `nas_RejectReasonTlv` `unpack_nas_SetEventReportInd_t::RRTlv`

8.852.2.4 `nas_SLQSSignalStrengthsTlv` `unpack_nas_SetEventReportInd_t::SLQSSSTlv`

8.852.2.5 `nas_SignalStrengthTlv` `unpack_nas_SetEventReportInd_t::SSTlv`

## 8.853 `unpack_nas_SetNasLTECphyCalndCallback_ind_t` Struct Reference

### Data Fields

- [nas\\_PhyCaAggScellIndType](#) `sPhyCaAggScellIndType`
- [nas\\_PhyCaAggScellDIBw](#) `sPhyCaAggScellDIBw`
- [nas\\_PhyCaAggScellInfo](#) `sPhyCaAggScellInfo`
- [nas\\_PhyCaAggPcellInfo](#) `sPhyCaAggPcellInfo`
- [nas\\_PhyCaAggScellIndex](#) `sPhyCaAggScellIndex`
- [nas\\_PhyCaAggDIBW](#) `sPhyCaAggDIBW`
- [nas\\_NumScellsConfig](#) `sNumScellsConfig`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.853.1 Detailed Description

Structure for storing the LTE PHY CA indication parameters.

#### Parameters

<i>pPhyCaAggScellIndType</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndType</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
------------------------------	--

<i>sPhyCaAgg- ScellDIBw</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellDIBw</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>sPhyCaAgg- ScellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>sPhyCaAgg- PcellInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggPcellInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>sPhyCaAgg- ScellIndex</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggScellIndex</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>sPhyCaAggDIB- W</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PhyCaAggDIBW</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>sNumScells- Config</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_NumScellsConfig</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.853.2 Field Documentation

8.853.2.1 `swi_uint256_t unpack_nas_SetNasLTECphyCalndCallback_ind_t::ParamPresenceMask`

8.853.2.2 `nas_NumScellsConfig unpack_nas_SetNasLTECphyCalndCallback_ind_t::sNumScellsConfig`

8.853.2.3 `nas_PhyCaAggDIBW unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggDIBW`

8.853.2.4 `nas_PhyCaAggPcellInfo unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggPcellInfo`

8.853.2.5 `nas_PhyCaAggScellDIBw unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellDIBw`

8.853.2.6 `nas_PhyCaAggScellIndex unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndex`

8.853.2.7 `nas_PhyCaAggScellIndType unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellIndType`

8.853.2.8 `nas_PhyCaAggScellInfo unpack_nas_SetNasLTECphyCalndCallback_ind_t::sPhyCaAggScellInfo`

## 8.854 unpack\_nas\_SetNetworkPreference\_t Struct Reference

### Data Fields

- `uint16_t Tlvresult`
- `swi_uint256_t ParamPresenceMask`

### 8.854.1 Detailed Description

This structure contains unpack set network preference parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>unpack result</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.854.2 Field Documentation

8.854.2.1 `swi_uint256_t unpack_nas_SetNetworkPreference_t::ParamPresenceMask`

8.854.2.2 `uint16_t unpack_nas_SetNetworkPreference_t::Tlvresult`

## 8.855 `unpack_nas_SetRoamingIndicatorCallback_ind_t` Struct Reference

#### Data Fields

- `uint8_t roaming`
- `swi_uint256_t ParamPresenceMask`

### 8.855.1 Detailed Description

This structure contains unpack set roaming indicator callback indication parameters.

#### 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> <li>Bit to check in ParamPresenceMask - 16</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.855.2 Field Documentation

8.855.2.1 `swi_uint256_t unpack_nas_SetRoamingIndicatorCallback_ind_t::ParamPresenceMask`

8.855.2.2 `uint8_t unpack_nas_SetRoamingIndicatorCallback_ind_t::roaming`

## 8.856 `unpack_nas_SetServingSystemCallback_ind_t` Struct Reference



## Data Fields

- [NASServingSystemInfo](#) SSInfo
- [uint16\\_t](#) Tlvresult
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.856.1 Detailed Description

This structure contains unpack set serving system callback indication parameters.

#### 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="#">NASServingSystemInfo</a> for more details</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.856.2 Field Documentation

8.856.2.1 [swi\\_uint256\\_t](#) unpack\_nas\_SetServingSystemCallback\_ind\_t::ParamPresenceMask

8.856.2.2 [NASServingSystemInfo](#) unpack\_nas\_SetServingSystemCallback\_ind\_t::SSInfo

8.856.2.3 [uint16\\_t](#) unpack\_nas\_SetServingSystemCallback\_ind\_t::Tlvresult

## 8.857 unpack\_nas\_SLQSGetErrorRate\_t Struct Reference

## Data Fields

- [uint16\\_t](#) \* [pCDMAFrameErrRate](#)
- [uint16\\_t](#) \* [pHDRPackErrRate](#)
- [uint8\\_t](#) \* [pGSMBER](#)
- [uint8\\_t](#) \* [pWCDMABER](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.857.1 Detailed Description

Structure for Error Rate unpack parameters.

## Parameters

<i>pCDMAFrameErrRate</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHDRPackErrRate</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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pGSMBER</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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pWCDMABER</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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.857.2 Field Documentation

8.857.2.1 swi\_uint256\_t unpack\_nas\_SLQSGetErrorRate\_t::ParamPresenceMask

8.857.2.2 uint16\_t\* unpack\_nas\_SLQSGetErrorRate\_t::pCDMAFrameErrRate

8.857.2.3 uint8\_t\* unpack\_nas\_SLQSGetErrorRate\_t::pGSMBER

8.857.2.4 uint16\_t\* unpack\_nas\_SLQSGetErrorRate\_t::pHDRPackErrRate

8.857.2.5 uint8\_t\* unpack\_nas\_SLQSGetErrorRate\_t::pWCDMABER

## 8.858 unpack\_nas\_SLQSGetHomeNetwork\_t Struct Reference

## Data Fields

- uint16\_t [mcc](#)
- uint16\_t [mnc](#)
- char [name](#) [255]
- uint16\_t [sid](#)
- uint16\_t [nid](#)
- [nas\\_homeNwMNC3GppTlv](#) \* [pHomeNwMNC3Gpp](#)

- [nas\\_nwNameSrc3GppTlv](#) \* [pNwNameSrc3Gpp](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.858.1 Detailed Description

This structure contains unpack get get home network parameters.

#### Parameters

<i>mcc</i>	<ul style="list-style-type: none"> <li>• Mobile country code (UMTS only).</li> <li>• A 16-bit integer representation of MCC. Range: 0 to 999.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>1</b></li> </ul>
<i>mnc</i>	<ul style="list-style-type: none"> <li>• Mobile network code (UMTS only).</li> <li>• A 16-bit integer representation of MNC. Range: 0 to 999.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>1</b></li> </ul>
<i>name</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>1</b></li> </ul>
<i>sid</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>nid</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pHomeNwMNC3Gpp[OUT]</i>	<ul style="list-style-type: none"> <li>• 3GPP Home Network MNC</li> <li>• See <a href="#">nas_homeNwMNC3GppTlv</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pNwNameSrc3Gpp[OUT]</i>	<ul style="list-style-type: none"> <li>• 3GPP Network Name Source</li> <li>• See <a href="#">nas_nwNameSrc3GppTlv</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.858.2 Field Documentation

- 8.858.2.1 uint16\_t unpack\_nas\_SLQSGetHomeNetwork\_t::mcc
- 8.858.2.2 uint16\_t unpack\_nas\_SLQSGetHomeNetwork\_t::mnc
- 8.858.2.3 char unpack\_nas\_SLQSGetHomeNetwork\_t::name[255]
- 8.858.2.4 uint16\_t unpack\_nas\_SLQSGetHomeNetwork\_t::nid
- 8.858.2.5 swi\_uint256\_t unpack\_nas\_SLQSGetHomeNetwork\_t::ParamPresenceMask
- 8.858.2.6 nas\_homeNwMNC3GppTlv\* unpack\_nas\_SLQSGetHomeNetwork\_t::pHomeNwMNC3Gpp
- 8.858.2.7 nas\_nwNameSrc3GppTlv\* unpack\_nas\_SLQSGetHomeNetwork\_t::pNwNameSrc3Gpp
- 8.858.2.8 uint16\_t unpack\_nas\_SLQSGetHomeNetwork\_t::sid

## 8.859 unpack\_nas\_SlqsGetLTECphyCAInfo\_t Struct Reference

### Data Fields

- [NasGetLTECphyCAInfo](#) [LTECphyCAInfo](#)
- uint16\_t [Tlvresult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.859.1 Detailed Description

This structure contains unpack get carrier aggregation event information parameters.

#### 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> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.859.2 Field Documentation

- 8.859.2.1 [NasGetLTECphyCAInfo](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::LTECphyCAInfo](#)
- 8.859.2.2 [swi\\_uint256\\_t](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::ParamPresenceMask](#)
- 8.859.2.3 [uint16\\_t](#) [unpack\\_nas\\_SlqsGetLTECphyCAInfo\\_t::Tlvresult](#)

## 8.860 unpack\_nas\_SLQSGetNetworkTime\_t Struct Reference

## Data Fields

- [nas\\_timeInfo](#) \* [p3GPP2TimeInfo](#)
- [nas\\_timeInfo](#) \* [p3GPPTimeInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.860.1 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>
------------------------	--

- \* - Bit to check in ParamPresenceMask - **16**

#### Parameters

<i>p3GPPTimeInfo</i>	[Optional] <ul style="list-style-type: none"> <li>– See <a href="#">nas_timeInfo</a> for more information</li> </ul>
----------------------	--

- \* - Bit to check in ParamPresenceMask - **17**

#### Parameters

<i>ParamPresence-Mask</i>	– bitmask representation to indicate valid parameters.
---------------------------	--

### 8.860.2 Field Documentation

8.860.2.1 [nas\\_timeInfo](#)\* [unpack\\_nas\\_SLQSGetNetworkTime\\_t::p3GPP2TimeInfo](#)

8.860.2.2 [nas\\_timeInfo](#)\* [unpack\\_nas\\_SLQSGetNetworkTime\\_t::p3GPPTimeInfo](#)

8.860.2.3 [swi\\_uint256\\_t](#) [unpack\\_nas\\_SLQSGetNetworkTime\\_t::ParamPresenceMask](#)

## 8.861 unpack\_nas\_SLQSGetOperatorNameData\_t Struct Reference

## Data Fields

- [nas\\_serviceProviderName](#) \* [pSrvcProviderName](#)
- [nas\\_operatorPLMNList](#) \* [pOperatorPLMNList](#)
- [nas\\_PLMNNetworkName](#) \* [pPLMNNetworkName](#)
- [nas\\_operatorNameString](#) \* [pOperatorNameString](#)
- [nas\\_PLMNNetworkNameData](#) \* [pNITZInformation](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.861.1 Detailed Description

Structure for Operator Name unpack.

## Parameters

<i>pSvcProvider-Name</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">nas_serviceProviderName</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pOperatorPLM-List</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">nas_operatorPLMNList</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pPLMNNetwork-Name</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">nas_PLMNNetworkName</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pOperatorName-String</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">nas_operatorNameString</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pNITZ-Information</i>	<ul style="list-style-type: none"> <li>Refer <a href="#">nas_PLMNNetworkNameData</a> for details (Optional).</li> <li>Can provide NULL if this parameter is not required.</li> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.861.2 Field Documentation

8.861.2.1 `swi_uint256_t unpack_nas_SLQSGetOperatorNameData_t::ParamPresenceMask`

8.861.2.2 `nas_PLMNNetworkNameData* unpack_nas_SLQSGetOperatorNameData_t::pNITZInformation`

8.861.2.3 `nas_operatorNameString* unpack_nas_SLQSGetOperatorNameData_t::pOperatorNameString`

8.861.2.4 `nas_operatorPLMNList* unpack_nas_SLQSGetOperatorNameData_t::pOperatorPLMNList`

8.861.2.5 `nas_PLMNNetworkName* unpack_nas_SLQSGetOperatorNameData_t::pPLMNNetworkName`

8.861.2.6 `nas_serviceProviderName* unpack_nas_SLQSGetOperatorNameData_t::pSvcProviderName`

8.862 `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]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.862.1 Detailed Description

This structure contains unpack get operator name for specified network 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> <li>– Note: This value is ignored if spnLength is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>spnLength</i>	<ul style="list-style-type: none"> <li>• Length of SPN which follows</li> <li>• Note: This SPN value is ignored if spnLength is zero and spnEncoding is 0xff.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>spn</i>	<ul style="list-style-type: none"> <li>• Service Provider name string</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>shortNameLen</i>	<ul style="list-style-type: none"> <li>• Length of shortName which follows</li> <li>• Note: This shortName value is ignored if shortNameLen is zero and shortNameEn is 0xff.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>shortName</i>	<ul style="list-style-type: none"> <li>• PLMN short name</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>– Note: This value is ignored if shortNameLen is zero.</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>longNameLen</i>	<ul style="list-style-type: none"> <li>• Length of longName which follows</li> <li>• Note: This longName value is ignored if longNameLen is zero and longNameEn is 0xff.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>longName</i>	<ul style="list-style-type: none"> <li>• PLMN long name</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.862.2 Field Documentation

8.862.2.1 char unpack\_nas\_SLQSGetPLMNName\_t::longName[255]

8.862.2.2 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameCI

8.862.2.3 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameEn

8.862.2.4 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameLen

8.862.2.5 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::longNameSB

8.862.2.6 swi\_uint256\_t unpack\_nas\_SLQSGetPLMNName\_t::ParamPresenceMask

8.862.2.7 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortName[255]

8.862.2.8 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameCI

8.862.2.9 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameEn

8.862.2.10 char unpack\_nas\_SLQSGetPLMNName\_t::shortNameLen

8.862.2.11 uint8\_t unpack\_nas\_SLQSGetPLMNName\_t::shortNameSB

8.862.2.12 char unpack\_nas\_SLQSGetPLMNName\_t::spn[255]

8.862.2.13 `uint8_t unpack_nas_SLQSGetPLMNName_t::spnEncoding`

8.862.2.14 `uint8_t unpack_nas_SLQSGetPLMNName_t::spnLength`

## 8.863 `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`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.863.1 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="#">nas_servSystem</a> for more information</li> <li>• Bit to check in <code>ParamPresenceMask</code> - 1</li> </ul>
----------------------	---

<i>RoamIndicator-Val</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>DataSrv-Capabilities</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Data services capability</li> <li>• See <a href="#">nas_dataSrvCapabilities</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>CurrentPLMN</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Current PLMN</li> <li>• See <a href="#">nas_currentPLMN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>SystemID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating System ID</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>NetworkID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Network ID</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>BaseStationID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Base Station Identification Number</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>Roaming-IndicatorList</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Roaming Indicator List</li> <li>• See <a href="#">nas_roamIndList</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></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> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>Gpp2TimeZone</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating 3GPP2 Time Zone</li> <li>• See <a href="#">nas_qaQmi3Gpp2TimeZone</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>CDMA_P_Rev</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating CDMA P_Rev in use</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></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> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>Lac</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating 3GPP Location Area Code</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>CellID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating 3GPP Cell ID</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>30</b></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> <li>• Bit to check in ParamPresenceMask - <b>31</b></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> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>DetailedSvcInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Detailed service information</li> <li>• See <a href="#">nas_detailSvcInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>33</b></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="#">nas_CDMASysInfoExt</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>TrackAreaCode</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Tracking area code information for LTE</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>CallBarStatus</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Call Barring Status</li> <li>• See <a href="#">nas_callBarStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.863.2 Field Documentation

8.863.2.1 `uint16_t unpack_nas_SLQSGetServingSystem_t::BasestationID`

8.863.2.2 `uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLatitude`

8.863.2.3 `uint32_t unpack_nas_SLQSGetServingSystem_t::BasestationLongitude`

8.863.2.4 `nas_callBarStatus unpack_nas_SLQSGetServingSystem_t::CallBarStatus`

8.863.2.5 `uint8_t unpack_nas_SLQSGetServingSystem_t::CDMA_P_Rev`

8.863.2.6 `nas_CDMASysInfoExt unpack_nas_SLQSGetServingSystem_t::CDMASystemInfoExt`

- 8.863.2.7 uint32\_t unpack\_nas\_SLQSGetServingSystem\_t::CellID
- 8.863.2.8 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::ConcSvcInfo
- 8.863.2.9 nas\_currentPLMN unpack\_nas\_SLQSGetServingSystem\_t::CurrentPLMN
- 8.863.2.10 nas\_dataSrvCapabilities unpack\_nas\_SLQSGetServingSystem\_t::DataSrvCapabilities
- 8.863.2.11 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::DefaultRoamInd
- 8.863.2.12 nas\_detailSvcInfo unpack\_nas\_SLQSGetServingSystem\_t::DetailedSvcInfo
- 8.863.2.13 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::DTMInd
- 8.863.2.14 nas\_qaQmi3Gpp2TimeZone unpack\_nas\_SLQSGetServingSystem\_t::Gpp2TimeZone
- 8.863.2.15 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::GppNetworkDSTAdjustment
- 8.863.2.16 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::GppTimeZone
- 8.863.2.17 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::HdrPersonality
- 8.863.2.18 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::Lac
- 8.863.2.19 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::NetworkID
- 8.863.2.20 swi\_uint256\_t unpack\_nas\_SLQSGetServingSystem\_t::ParamPresenceMask
- 8.863.2.21 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::PRLInd
- 8.863.2.22 uint8\_t unpack\_nas\_SLQSGetServingSystem\_t::RoamIndicatorVal
- 8.863.2.23 nas\_roamIndList unpack\_nas\_SLQSGetServingSystem\_t::RoamingIndicatorList
- 8.863.2.24 nas\_servSystem unpack\_nas\_SLQSGetServingSystem\_t::ServingSystem
- 8.863.2.25 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::SystemID
- 8.863.2.26 uint16\_t unpack\_nas\_SLQSGetServingSystem\_t::TrackAreaCode

## 8.864 unpack\_nas\_SLQSGetServingSystemV2\_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 DTMLnd](#)
- [nas\\_detailSvcInfo](#) DetailedSvcInfo
- [nas\\_CDMA SysInfoExt](#) CDMA SystemInfoExt
- [uint8\\_t HdrPersonality](#)
- [uint16\\_t TrackAreaCode](#)
- [nas\\_callBarStatus](#) CallBarStatus
- [nas\\_MNCPCSDigitStatus](#) MNCPCSDigitStatTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.864.1 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="#">nas_servSystem</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>RoamIndicator-Val</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>DataSrv-Capabilities</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Data services capability</li> <li>• See <a href="#">nas_dataSrvCapabilities</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>CurrentPLMN</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Current PLMN</li> <li>• See <a href="#">nas_currentPLMN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>SystemID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating System ID</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>NetworkID</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Network ID</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>BaseStationID</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Base Station Identification Number</li> <li>Bit to check in ParamPresenceMask - <b>20</b></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> <li>Bit to check in ParamPresenceMask - <b>20</b></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> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>Roaming-IndicatorList</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating Roaming Indicator List</li> <li>See <a href="#">nas_roamIndList</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>21</b></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> <li>Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>Gpp2TimeZone</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating 3GPP2 Time Zone</li> <li>See <a href="#">nas_qaQmi3Gpp2TimeZone</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>CDMA_P_Rev</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating CDMA P_Rev in use</li> <li>Bit to check in ParamPresenceMask - <b>24</b></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> <li>Bit to check in ParamPresenceMask - <b>26</b></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> <li>Bit to check in ParamPresenceMask - <b>27</b></li> </ul>



<i>Lac</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating 3GPP Location Area Code</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>CellID</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating 3GPP Cell ID</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>30</b></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> <li>• Bit to check in ParamPresenceMask - <b>31</b></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> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>DetailedSvcInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Detailed service information</li> <li>• See <a href="#">nas_detailSvcInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>33</b></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="#">nas_CDMASysInfoExt</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>TrackAreaCode</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Tracking area code information for LTE</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>

<i>CallBarStatus</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Call Barring Status</li> <li>• See <a href="#">nas_callBarStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>MNCPCSDigit-Tlv</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating MNC PCS Digit Include Status</li> <li>• See <a href="#">nas_MNCPCSDigitStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.864.2 Field Documentation

- 8.864.2.1 `uint16_t unpack_nas_SLQSGetservingSystemV2_t::BasestationID`
- 8.864.2.2 `uint32_t unpack_nas_SLQSGetservingSystemV2_t::BasestationLatitude`
- 8.864.2.3 `uint32_t unpack_nas_SLQSGetservingSystemV2_t::BasestationLongitude`
- 8.864.2.4 `nas_callBarStatus unpack_nas_SLQSGetservingSystemV2_t::CallBarStatus`
- 8.864.2.5 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::CDMA_P_Rev`
- 8.864.2.6 `nas_CDMA_SysInfoExt unpack_nas_SLQSGetservingSystemV2_t::CDMA_SystemInfoExt`
- 8.864.2.7 `uint32_t unpack_nas_SLQSGetservingSystemV2_t::CellID`
- 8.864.2.8 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::ConcSvcInfo`
- 8.864.2.9 `nas_currentPLMN unpack_nas_SLQSGetservingSystemV2_t::CurrentPLMN`
- 8.864.2.10 `nas_dataSrvCapabilities unpack_nas_SLQSGetservingSystemV2_t::DataSrvCapabilities`
- 8.864.2.11 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::DefaultRoamInd`
- 8.864.2.12 `nas_detailSvcInfo unpack_nas_SLQSGetservingSystemV2_t::DetailedSvcInfo`
- 8.864.2.13 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::DTMInd`
- 8.864.2.14 `nas_qaQmi3Gpp2TimeZone unpack_nas_SLQSGetservingSystemV2_t::Gpp2TimeZone`
- 8.864.2.15 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::GppNetworkDSTAdjustment`
- 8.864.2.16 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::GppTimeZone`
- 8.864.2.17 `uint8_t unpack_nas_SLQSGetservingSystemV2_t::HdrPersonality`
- 8.864.2.18 `uint16_t unpack_nas_SLQSGetservingSystemV2_t::Lac`
- 8.864.2.19 `nas_MNCPCSDigitStatus unpack_nas_SLQSGetservingSystemV2_t::MNCPCSDigitStatTlv`

- 8.864.2.20 uint16\_t unpack\_nas\_SLQSGetServingSystemV2\_t::NetworkID
- 8.864.2.21 swi\_uint256\_t unpack\_nas\_SLQSGetServingSystemV2\_t::ParamPresenceMask
- 8.864.2.22 uint8\_t unpack\_nas\_SLQSGetServingSystemV2\_t::PRLInd
- 8.864.2.23 uint8\_t unpack\_nas\_SLQSGetServingSystemV2\_t::RoamIndicatorVal
- 8.864.2.24 nas\_roamIndList unpack\_nas\_SLQSGetServingSystemV2\_t::RoamingIndicatorList
- 8.864.2.25 nas\_servSystem unpack\_nas\_SLQSGetServingSystemV2\_t::ServingSystem
- 8.864.2.26 uint16\_t unpack\_nas\_SLQSGetServingSystemV2\_t::SystemID
- 8.864.2.27 uint16\_t unpack\_nas\_SLQSGetServingSystemV2\_t::TrackAreaCode

## 8.865 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](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.865.1 Detailed Description

This structure contains the Signal Strength Information

## Parameters

<i>signalStrength-ReqMask</i>	<ul style="list-style-type: none"> <li>Request Mask           <ul style="list-style-type: none"> <li>Request additional signal information for: Bit 0 - RSSI Information bit Valid values are: 0 - Do Not Request Additional Info for RSSI 1 - Request Additional Info for RSSI</li> <li>Bit 1 - ECIO Information bit Valid values are: 0 - Do Not Request Additional Info for ECIO 1 - Request Additional Info for ECIO</li> <li>Bit 2 - IO Information bit Valid values are: 0 - Do Not Request Additional Info for IO 1 - Request Additional Info for IO</li> <li>Bit 3 - SINR Information bit Valid values are: 0 - Do Not Request Additional Info for SINR 1 - Request Additional Info for SINR</li> <li>Bit 4 - ERROR RATE Information bit Valid values are: 0 - Do Not Request Additional Info for Error Rate 1 - Request Additional Info for Error Rate</li> <li>Bit 5 - RSRQ Information bit Valid values are: 0 - Do Not Request Additional Info for RSRQ 1 - Request Additional Info for RSRQ</li> <li>Bit 6 - LTE SNR information bit Valid values are: 0 - Do not request additional information for LTE SNR 1 - Request additional information for LTE SNR</li> <li>Bit 7 - LTE RSRP Information bit Valid values are: 0 - Do not request additional information for LTE RSRP 1 - Request additional information for LTE RSRP</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>rxSignal-StrengthListLen</i>	<ul style="list-style-type: none"> <li>Number of elements in Receive Signal Strength List</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>rxSignal-StrengthList</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_rxSignalStrengthListElement</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ecioListLen</i>	<ul style="list-style-type: none"> <li>Number of elements in ECIO List</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ecioList</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_ecioListElement</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>io</i>	<ul style="list-style-type: none"> <li>Received Io in dBm; IO is only applicable for 1xEV-DO</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>sinr</i>	<ul style="list-style-type: none"> <li>SINR level <ul style="list-style-type: none"> <li>SINR is only applicable for 1xEV-DO; valid levels are 0 to 8 where maximum value for 0 - SINR_LEVEL_0 is -9 dB 1 - SINR_LEVEL_1 is -6 dB 2 - SINR_LEVEL_2 is -4.5 dB 3 - SINR_LEVEL_3 is -3 dB 4 - SINR_LEVEL_4 is -2 dB 5 - SINR_LEVEL_5 is +1 dB 6 - SINR_LEVEL_6 is +3 dB 7 - SINR_LEVEL_7 is +6 dB 8 - SINR_LEVEL_8 is +9 dB</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>errorRateListLen</i>	<ul style="list-style-type: none"> <li>Number of elements in Error Rate List</li> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>errorRateList</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_errorRateListElement</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>rsrqInfo</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_rsrqInformation</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ltesnr</i>	<ul style="list-style-type: none"> <li>LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE</li> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>ltersrp</i>	<ul style="list-style-type: none"> <li>LTE SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246. LTE SNR is included only when the current serving system is LTE</li> <li>Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.865.2 Field Documentation

8.865.2.1 `nas_ecioListElement` `unpack_nas_SLQSGetSignalStrength_t::ecioList[18]`

8.865.2.2 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::ecioListLen`

8.865.2.3 `nas_errorRateListElement` `unpack_nas_SLQSGetSignalStrength_t::errorRateList[18]`

8.865.2.4 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::errorRateListLen`

8.865.2.5 `int32_t` `unpack_nas_SLQSGetSignalStrength_t::lo`

8.865.2.6 `int16_t` `unpack_nas_SLQSGetSignalStrength_t::ltersrp`

8.865.2.7 `int16_t` `unpack_nas_SLQSGetSignalStrength_t::ltesnr`

8.865.2.8 `swi_uint256_t` `unpack_nas_SLQSGetSignalStrength_t::ParamPresenceMask`

8.865.2.9 `nas_rsrqInformation` `unpack_nas_SLQSGetSignalStrength_t::rsrqInfo`

8.865.2.10 `nas_rxSignalStrengthListElement` `unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthList[18]`

8.865.2.11 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::rxSignalStrengthListLen`

8.865.2.12 `uint16_t` `unpack_nas_SLQSGetSignalStrength_t::signalStrengthReqMask`

8.865.2.13 `uint8_t` `unpack_nas_SLQSGetSignalStrength_t::sinr`

## 8.866 `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`
- `uint32_t` \* `pCampedExceptionMode`
- `swi_uint256_t` `ParamPresenceMask`

### 8.866.1 Detailed Description

This structure contains unpack get system information parameters.

#### Parameters

<i>pCDMASrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHDRSrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pGSMSrvStatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>pWCDMASrv- StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pLTESrvStatus- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_HDRSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pWCDMASys- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTESysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pAddCDMASys- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pAddGSMSys- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pAddWCDMA- SysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>

<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pCampedCiot-LteOpMode</i>	<ul style="list-style-type: none"> <li>• Camped CIOT LTE Operational Mode.</li> <li>• Values <ul style="list-style-type: none"> <li>– NAS_CIOT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIOT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIOT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIOT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>73</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.866.2 Field Documentation

8.866.2.1 `nas_AddCDMASysInfo* unpack_nas_SLQSGetSysInfo_t::pAddCDMASysInfo`

8.866.2.2 `nas_AddSysInfo* unpack_nas_SLQSGetSysInfo_t::pAddGSMSysInfo`

8.866.2.3 `uint16_t* unpack_nas_SLQSGetSysInfo_t::pAddHDRSysInfo`



- 8.866.2.4 uint16\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pAddLTESysInfo
- 8.866.2.5 nas\_AddSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pAddWCDMASysInfo
- 8.866.2.6 swi\_uint256\_t unpack\_nas\_SLQSGetSysInfo\_t::ParamPresenceMask
- 8.866.2.7 uint32\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pCampedCiotLteOpMode
- 8.866.2.8 nas\_SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pCDMASrvStatusInfo
- 8.866.2.9 nas\_CDMASysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pCDMASysInfo
- 8.866.2.10 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSMCallBarringSysInfo
- 8.866.2.11 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pGSMCipherDomainSysInfo
- 8.866.2.12 nas\_GSMSrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSMSrvStatusInfo
- 8.866.2.13 nas\_GSMSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pGSMSysInfo
- 8.866.2.14 nas\_SrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDRSrvStatusInfo
- 8.866.2.15 nas\_HDRSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pHDRSysInfo
- 8.866.2.16 nas\_GSMSrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTESrvStatusInfo
- 8.866.2.17 nas\_LTESysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pLTESysInfo
- 8.866.2.18 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pLTEVoiceSupportSysInfo
- 8.866.2.19 nas\_CallBarringSysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMACallBarringSysInfo
- 8.866.2.20 uint8\_t\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMACipherDomainSysInfo
- 8.866.2.21 nas\_GSMSrvStatusInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMASrvStatusInfo
- 8.866.2.22 nas\_WCDMASysInfo\* unpack\_nas\_SLQSGetSysInfo\_t::pWCDMASysInfo

## 8.867 unpack\_nas\_SLQSGetSysInfoV2\_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](#)
- [uint32\\_t](#) \* [pCampedExceptionLteOpMode](#)
- [nas\\_LteEmbmsCoverageTlv](#) \* [pLteEmbmsCoverage](#)
- [nas\\_SimRejInfoTlv](#) \* [pSimRejInfo](#)
- [nas\\_LmsVoiceSupportLteTlv](#) \* [pLmsVoiceSupportLte](#)
- [nas\\_LteVoiceDomainTlv](#) \* [pLteVoiceDomain](#)
- [nas\\_SrvRegRestrictionTlv](#) \* [pSrvRegRestriction](#)
- [nas\\_LteRegDomainTlv](#) \* [pLteRegDomain](#)
- [nas\\_LteEmbmsTraceIdTlv](#) \* [pLteEmbmsTraceId](#)
- [nas\\_NR5GSrvStatusTlv](#) \* [pNR5GSrvStatusInfo](#)
- [nas\\_NR5GSysInfoTlv](#) \* [pNR5GSysInfo](#)
- [nas\\_NR5GCellStatusInfoTlv](#) \* [pNR5GCellStatusInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.867.1 Detailed Description

This structure contains unpack get system information parameters.

#### Parameters

<i>pCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pHDRSrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pGSMSrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pWCDMASrv-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></li> </ul>
<i>pLTESrvStatus-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>20</b></li> </ul>
<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMASysInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>21</b></li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_HDRSysInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>22</b></li> </ul>

<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pWCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTESysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pAddCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pAddGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pGSMCallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pWCDMACallBarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>• Bit to check in ParamPresenceMask - <b>33</b></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> <li>• Bit to check in ParamPresenceMask - <b>34</b></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> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pCampedCiot-LteOpMode</i>	<ul style="list-style-type: none"> <li>• Camped CIOT LTE Operational Mode.</li> <li>• Values <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>73</b></li> </ul>
<i>pLteEmbms-Coverage</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteEmbmsCoverageTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pSimRejInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SimRejInfoTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>plmsVoice-SupportLte</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_lmsVoiceSupportLteTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>pLteVoice-Domain</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteVoiceDomainTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>
<i>pSrvReg-Restriction</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvRegRestrictionTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>47</b></li> </ul>
<i>pLteRegDomain</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteRegDomainTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>49</b></li> </ul>

<i>pLteEmbms-TraceId</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LteEmbmsTraceIdTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>52</b></li> </ul>
<i>pNR5GSrv-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_NR5GSrvStatusTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>74</b></li> </ul>
<i>pNR5gSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_NR5GSysInfoTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>75</b></li> </ul>
<i>pNR5GCell-StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_NR5GCellStatusInfoTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>76</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.867.2 Field Documentation

8.867.2.1 **nas\_AddCDMASysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pAddCDMASysInfo

8.867.2.2 **nas\_AddSysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pAddGSM SysInfo

8.867.2.3 **uint16\_t\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pAddHDRSysInfo

8.867.2.4 **uint16\_t\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pAddLTESysInfo

8.867.2.5 **nas\_AddSysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pAddWCDMASysInfo

8.867.2.6 **swi\_uint256\_t** unpack\_nas\_SLQSGetSysInfoV2\_t::ParamPresenceMask

8.867.2.7 **uint32\_t\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pCampedCiotLteOpMode

8.867.2.8 **nas\_SrvStatusInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pCDMASrvStatusInfo

8.867.2.9 **nas\_CDMASysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pCDMASysInfo

8.867.2.10 **nas\_CallBarringSysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pGSMCallBarringSysInfo

8.867.2.11 **uint8\_t\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pGSMCipherDomainSysInfo

8.867.2.12 **nas\_GSMSrvStatusInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pGSM SrvStatusInfo

8.867.2.13 **nas\_GSMSysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pGSM SysInfo

8.867.2.14 **nas\_SrvStatusInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pHDRSrvStatusInfo

8.867.2.15 **nas\_HDRSysInfo\*** unpack\_nas\_SLQSGetSysInfoV2\_t::pHDRSysInfo

8.867.2.16 **nas\_ImsVoiceSupportLteTlv\*** unpack\_nas\_SLQSGetSysInfoV2\_t::plmsVoiceSupportLte

8.867.2.17 `nas_LteEmbmsCoverageTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pLteEmbmsCoverage`

8.867.2.18 `nas_LteEmbmsTraceIdTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pLteEmbmsTraceId`

8.867.2.19 `nas_LteRegDomainTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pLteRegDomain`

8.867.2.20 `nas_GSMsSrvStatusInfo*` `unpack_nas_SLQSGetSysInfoV2_t::pLTeSrvStatusInfo`

8.867.2.21 `nas_LTESysInfo*` `unpack_nas_SLQSGetSysInfoV2_t::pLTeSysInfo`

8.867.2.22 `nas_LteVoiceDomainTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pLteVoiceDomain`

8.867.2.23 `uint8_t*` `unpack_nas_SLQSGetSysInfoV2_t::pLTeVoiceSupportSysInfo`

8.867.2.24 `nas_NR5GCellStatusInfoTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pNR5GCellStatusInfo`

8.867.2.25 `nas_NR5GSrvStatusTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pNR5GSrvStatusInfo`

8.867.2.26 `nas_NR5GSysInfoTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pNR5GSysInfo`

8.867.2.27 `nas_SimRejInfoTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pSimRejInfo`

8.867.2.28 `nas_SrvRegRestrictionTlv*` `unpack_nas_SLQSGetSysInfoV2_t::pSrvRegRestriction`

8.867.2.29 `nas_CallBarringSysInfo*` `unpack_nas_SLQSGetSysInfoV2_t::pWCDMA_CallBarringSysInfo`

8.867.2.30 `uint8_t*` `unpack_nas_SLQSGetSysInfoV2_t::pWCDMACipherDomainSysInfo`

8.867.2.31 `nas_GSMsSrvStatusInfo*` `unpack_nas_SLQSGetSysInfoV2_t::pWCDMASrvStatusInfo`

8.867.2.32 `nas_WCDMASysInfo*` `unpack_nas_SLQSGetSysInfoV2_t::pWCDMASysInfo`

## 8.868 `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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.868.1 Detailed Description

This structure contains unpack get system selection preferences parameters.

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

- \* - Bit to check in ParamPresenceMask - **16**

## Parameters

<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>
------------------	--

- \* - Bit to check in ParamPresenceMask - **17**

## Parameters

<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>
------------------	--

- \* - Bit to check in ParamPresenceMask - 18



## Parameters

<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>
-----------------	---

- \* - Bit to check in ParamPresenceMask - **19**

## Parameters

<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>
------------------	---

- \* - Bit to check in ParamPresenceMask - **20**

## Parameters

<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>
---------------------	--

- \* - Bit to check in ParamPresenceMask - **21**

#### Parameters

<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>
--------------------	---

- \* - Bit to check in ParamPresenceMask - **22**

## Parameters

<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>
-----------------------	--

- \* - Bit to check in ParamPresenceMask - 24

## Parameters

<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>
-------------------------	---

- \* - Bit to check in ParamPresenceMask - 25

## Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>– bitmask representation to indicate valid parameters.</li> </ul>
---------------------------	--

## 8.868.2 Field Documentation

8.868.2.1 swi\_uint256\_t unpack\_nas\_SLQSGetSysSelectionPref\_t::ParamPresenceMask

8.868.2.2 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pBandPref

8.868.2.3 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pEmerMode

8.868.2.4 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pGWAcqOrderPref

8.868.2.5 uint64\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pLTEBandPref

8.868.2.6 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pModePref

8.868.2.7 uint8\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pNetSelPref

8.868.2.8 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pPRLPref

8.868.2.9 uint16\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pRoamPref

8.868.2.10 uint32\_t\* unpack\_nas\_SLQSGetSysSelectionPref\_t::pSrvDomainPref

## 8.869 unpack\_nas\_SLQSGetSysSelectionPrefExt\_t Struct Reference

## Data Fields

- [nas\\_EmerModeTlv](#) \* [pEmerMode](#)
- [nas\\_ModePrefTlv](#) \* [pModePref](#)
- [nas\\_BandPrefTlv](#) \* [pBandPref](#)
- [nas\\_PRLPrefTlv](#) \* [pPRLPref](#)
- [nas\\_RoamPrefTlv](#) \* [pRoamPref](#)
- [nas\\_LTEBandPrefTlv](#) \* [pLTEBandPref](#)
- [nas\\_NetSelPrefTlv](#) \* [pNetSelPref](#)
- [nas\\_SrvDomainPrefTlv](#) \* [pSrvDomainPref](#)
- [nas\\_GWAcqOrderPrefTlv](#) \* [pGWAcqOrderPref](#)
- [nas\\_AcqOrderPrefTlv](#) \* [pAcqOrderPref](#)
- [nas\\_RatDisabledMaskTlv](#) \* [pRatDisabledMask](#)
- [nas\\_CiotLteOpModePrefTlv](#) \* [pCiotLteOpModePref](#)
- [nas\\_LteM1BandPrefTlv](#) \* [pLteM1BandPref](#)
- [nas\\_LteNb1BandPrefTlv](#) \* [pLteNb1BandPref](#)
- [nas\\_CiotAcqOrderPrefTlv](#) \* [pCiotAcqOrderPref](#)
- [nas\\_NR5gBandPrefTlv](#) \* [pNr5gBandPref](#)
- [nas\\_LTEBandPrefExtTlv](#) \* [pLTEBandPrefExt](#)

### 8.869.1 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>See <a href="#">nas_EmerModeTlv</a> for more information</li> </ul>
<i>pModePref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the radio technology mode preference</li> <li>See <a href="#">nas_ModePrefTlv</a> for more information</li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter representing the band preference</li> <li>See <a href="#">nas_BandPrefTlv</a> for more information</li> </ul>
<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the CDMA PRL Preference</li> <li>See <a href="#">nas_PRLPrefTlv</a> for more information</li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating the roaming Preference</li> <li>See <a href="#">nas_RoamPrefTlv</a> for more information</li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>Optional parameter representing the LTE band preference</li> <li>See <a href="#">nas_LTEBandPrefTlv</a> for more information</li> </ul>
<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>Optional parameter indicating network selection preference</li> <li>See <a href="#">nas_NetSelPrefTlv</a> for more information</li> </ul>

<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• See <a href="#">nas_SrvDomainPrefTlv</a> for more information</li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• See <a href="#">nas_GWAcqOrderPrefTlv</a> for more information</li> </ul>
<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Acquisition Order Preference</li> <li>• See <a href="#">nas_AcqOrderPrefTlv</a> for more information</li> </ul>
<i>pRatDisabled-Mask</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the radio technologies that are disabled.</li> <li>• See <a href="#">nas_RatDisabledMaskTlv</a> for more information</li> </ul>
<i>pCiotLteOp-ModePref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• CIOT LTE Operational Mode Preference</li> <li>• See <a href="#">nas_CiotLteOpModePrefTlv</a> for more information</li> </ul>
<i>pLteM1BandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the LTE M1 band preference</li> <li>• See <a href="#">nas_LteM1BandPrefTlv</a> for more information</li> </ul>
<i>pLteNb1Band-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the LTE NB1 band preference.</li> <li>• See <a href="#">nas_LteNb1BandPrefTlv</a> for more information</li> </ul>
<i>pCiotAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating CIOT Acquisition Order Preference</li> <li>• See <a href="#">nas_CiotAcqOrderPrefTlv</a> for more information</li> </ul>
<i>pNr5gBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating NR5G Band Preference</li> <li>• See <a href="#">nas_NR5gBandPrefTlv</a> for more information</li> </ul>
<i>pLTEBandPref-Ext</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating LTE Band Preference Extended</li> <li>• See <a href="#">nas_LTEBandPrefExtTlv</a> for more information</li> </ul>

## 8.869.2 Field Documentation

8.869.2.1 `nas_AcqOrderPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pAcqOrderPref`

8.869.2.2 `nas_BandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pBandPref`

8.869.2.3 `nas_CiotAcqOrderPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotAcqOrderPref`

- 8.869.2.4 `nas_CiotLteOpModePrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pCiotLteOpModePref`
- 8.869.2.5 `nas_EmerModeTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pEmerMode`
- 8.869.2.6 `nas_GWAcqOrderPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pGWAcqOrderPref`
- 8.869.2.7 `nas_LTEBandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pLTEBandPref`
- 8.869.2.8 `nas_LTEBandPrefExtTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pLTEBandPrefExt`
- 8.869.2.9 `nas_LteM1BandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteM1BandPref`
- 8.869.2.10 `nas_LteNb1BandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pLteNb1BandPref`
- 8.869.2.11 `nas_ModePrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pModePref`
- 8.869.2.12 `nas_NetSelPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pNetSelPref`
- 8.869.2.13 `nas_NR5gBandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pNr5gBandPref`
- 8.869.2.14 `nas_PRLPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pPRLPref`
- 8.869.2.15 `nas_RatDisabledMaskTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pRatDisabledMask`
- 8.869.2.16 `nas_RoamPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pRoamPref`
- 8.869.2.17 `nas_SrvDomainPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExt_t::pSrvDomainPref`

## 8.870 `unpack_nas_SLQSGetSysSelectionPrefExtV2_t` Struct Reference

### Data Fields

- `nas_EmerModeTlv` \* `pEmerMode`
- `nas_ModePrefTlv` \* `pModePref`
- `nas_BandPrefTlv` \* `pBandPref`
- `nas_PRLPrefTlv` \* `pPRLPref`
- `nas_RoamPrefTlv` \* `pRoamPref`
- `nas_LTEBandPrefTlv` \* `pLTEBandPref`
- `nas_NetSelPrefTlv` \* `pNetSelPref`
- `nas_SrvDomainPrefTlv` \* `pSrvDomainPref`
- `nas_GWAcqOrderPrefTlv` \* `pGWAcqOrderPref`
- `nas_AcqOrderPrefTlv` \* `pAcqOrderPref`
- `nas_RatDisabledMaskTlv` \* `pRatDisabledMask`
- `nas_CiotLteOpModePrefTlv` \* `pCiotLteOpModePref`
- `nas_LteM1BandPrefTlv` \* `pLteM1BandPref`
- `nas_LteNb1BandPrefTlv` \* `pLteNb1BandPref`
- `nas_CiotAcqOrderPrefTlv` \* `pCiotAcqOrderPref`
- `nas_NR5gBandPrefTlv` \* `pNr5gBandPref`
- `nas_LTEBandPrefExtTlv` \* `pLTEBandPrefExt`
- `nas_TDSCDMABandPrefTlv` \* `pTDSCDMABandPref`
- `nas_SrvRegRestricTlv` \* `pSrvRegRestric`
- `nas_UsageSettingTlv` \* `pUsageSetting`
- `nas_VoiceDomainPrefTlv` \* `pVoiceDomainPref`
- `swi_uint256_t` `ParamPresenceMask`

### 8.870.1 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>• See <a href="#">nas_EmerModeTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pModePref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the radio technology mode preference</li> <li>• See <a href="#">nas_ModePrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the band preference</li> <li>• See <a href="#">nas_BandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pPRLPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the CDMA PRL Preference</li> <li>• See <a href="#">nas_PRLPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pRoamPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating the roaming Preference</li> <li>• See <a href="#">nas_RoamPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pLTEBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the LTE band preference</li> <li>• See <a href="#">nas_LTEBandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pNetSelPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating network selection preference</li> <li>• See <a href="#">nas_NetSelPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pSrvDomainPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Service domain preference</li> <li>• See <a href="#">nas_SrvDomainPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pGWAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating GSM/WCDMA Acquisition order Preference</li> <li>• See <a href="#">nas_GWAcqOrderPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>

<i>pAcqOrderPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• Acquisition Order Preference</li> <li>• See <a href="#">nas_AcqOrderPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pRatDisabled-Mask</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the radio technologies that are disabled.</li> <li>• See <a href="#">nas_RatDisabledMaskTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>pCiotLteOp-ModePref</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• CIOT LTE Operational Mode Preference</li> <li>• See <a href="#">nas_CiotLteOpModePrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>pLteM1BandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the LTE M1 band preference</li> <li>• See <a href="#">nas_LteM1BandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>pLteNb1Band-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter representing the LTE NB1 band preference.</li> <li>• See <a href="#">nas_LteNb1BandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>pCiotAcqOrder-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating CIOT Acquisition Order Preference</li> <li>• See <a href="#">nas_CiotAcqOrderPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>pNr5gBandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating NR5G Band Preference</li> <li>• See <a href="#">nas_NR5gBandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>pLTEBandPref-Ext</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating LTE Band Preference Extended</li> <li>• See <a href="#">nas_LTEBandPrefExtTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>pTDSCDMA-BandPref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating TDSCDMA Preference</li> <li>• See <a href="#">nas_TDSCDMABandPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pSrvRegRestric</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating SRV Registration Restriction</li> <li>• See <a href="#">nas_SrvRegRestricTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>



<i>pUsageSetting</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Usage settings</li> <li>• See <a href="#">nas_UsageSettingTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pVoiceDomain-Pref</i>	<ul style="list-style-type: none"> <li>• Optional parameter indicating Voice Domain Preference</li> <li>• See <a href="#">nas_VoiceDomainPrefTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.870.2 Field Documentation

8.870.2.1 **nas\_AcqOrderPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pAcqOrderPref

8.870.2.2 **swi\_uint256\_t** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::ParamPresenceMask

8.870.2.3 **nas\_BandPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pBandPref

8.870.2.4 **nas\_CiotAcqOrderPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pCiotAcqOrderPref

8.870.2.5 **nas\_CiotLteOpModePrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pCiotLteOpModePref

8.870.2.6 **nas\_EmerModeTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pEmerMode

8.870.2.7 **nas\_GWAcqOrderPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pGWAcqOrderPref

8.870.2.8 **nas\_LTEBandPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pLTEBandPref

8.870.2.9 **nas\_LTEBandPrefExtTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pLTEBandPrefExt

8.870.2.10 **nas\_LteM1BandPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pLteM1BandPref

8.870.2.11 **nas\_LteNb1BandPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pLteNb1BandPref

8.870.2.12 **nas\_ModePrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pModePref

8.870.2.13 **nas\_NetSelPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pNetSelPref

8.870.2.14 **nas\_NR5gBandPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pNr5gBandPref

8.870.2.15 **nas\_PRLPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pPRLPref

8.870.2.16 **nas\_RatDisabledMaskTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pRatDisabledMask

8.870.2.17 **nas\_RoamPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pRoamPref

8.870.2.18 **nas\_SrvDomainPrefTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pSrvDomainPref

8.870.2.19 **nas\_SrvRegRestricTlv\*** unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t::pSrvRegRestric

8.870.2.20 `nas_TDSCDMABandPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExtV2_t::pTDSCDMABandPref`

8.870.2.21 `nas_UsageSettingTlv*` `unpack_nas_SLQSGetSysSelectionPrefExtV2_t::pUsageSetting`

8.870.2.22 `nas_VoiceDomainPrefTlv*` `unpack_nas_SLQSGetSysSelectionPrefExtV2_t::pVoiceDomainPref`

## 8.871 `unpack_nas_SLQSNasEdrxChangeInfoCallback_Ind_t` Struct Reference

### Data Fields

- [nas\\_EdrxEnableType](#) `edrxEnableTypeTlv`
- [nas\\_EdrxCycleLength](#) `edrxCycleLengthTlv`
- [nas\\_EdrxPagingTimeWindow](#) `edrxPagingTimeWindowTlv`
- [nas\\_EdrxRatType](#) `edrxRatTypeTlv`
- [nas\\_EdrxCiotLteMode](#) `edrxCiotLteModeTlv`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.871.1 Detailed Description

Structure for storing the eDRX Change Info indication parameters.

#### Parameters

<i>edrxEnableType-Tlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_EdrxEnableType</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>edrxCycle-LengthTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_EdrxCycleLength</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>edrxPagingTime-WindowTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_EdrxPagingTimeWindow</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>edrxRatTypeTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_EdrxRatType</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>edrxCiotLte-ModeTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_EdrxCiotLteMode</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.871.2 Field Documentation

8.871.2.1 `nas_EdrxCiotLteMode` `unpack_nas_SLQSNasEdrxChangeInfoCallback_Ind_t::edrxCiotLteModeTlv`

8.871.2.2 `nas_EdrxCycleLength` `unpack_nas_SLQSNasEdrxChangeInfoCallback_Ind_t::edrxCycleLengthTlv`

8.871.2.3 `nas_EdrxEnableType` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxEnableTypeTlv`

8.871.2.4 `nas_EdrxPagingTimeWindow` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxPagingTimeWindow-Tlv`

8.871.2.5 `nas_EdrxRatType` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::edrxRatTypeTlv`

8.871.2.6 `swi_uint256_t` `unpack_nas_SLQSNasEdrxChangeInfoCallBack_Ind_t::ParamPresenceMask`

## 8.872 unpack\_nas\_SLQSNasGet3GPP2Subscription\_t Struct Reference

### Data Fields

- `nas_namName` \* `pNAMNameInfo`
- `nas_dirNum` \* `pDirNum`
- `nas_homeSIDNID` \* `pHomeSIDNID`
- `nas_minBasedIMSI` \* `pMinBasedIMSI`
- `nas_trueIMSI` \* `pTrueIMSI`
- `nas_CDMAChannel` \* `pCDMAChannel`
- `nas_Mdn` \* `pMdn`
- `swi_uint256_t` `ParamPresenceMask`

### 8.872.1 Detailed Description

Structure for 3GPP2Subscription unpack.

#### Parameters

<i>pNAMNameInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_namName</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pDirNum</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_dirNum</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pHomeSIDNID</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_homeSIDNID</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pMinBasedIMSI</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_minBasedIMSI</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pTrueIMSI</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_trueIMSI</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pCDMAChannel</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMAChannel</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>

<i>pMdn</i>	[Optional] <ul style="list-style-type: none"> <li>• See nas_mdn for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.872.2 Field Documentation

8.872.2.1 `swi_uint256_t` `unpack_nas_SLQSNasGet3GPP2Subscription_t::ParamPresenceMask`

8.872.2.2 `nas_CDMAChannel*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pCDMAChannel`

8.872.2.3 `nas_dirNum*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pDirNum`

8.872.2.4 `nas_homeSIDNID*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pHomeSIDNID`

8.872.2.5 `nas_Mdn*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pMdn`

8.872.2.6 `nas_minBasedIMSI*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pMinBasedIMSI`

8.872.2.7 `nas_namName*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pNAMNameInfo`

8.872.2.8 `nas_trueIMSI*` `unpack_nas_SLQSNasGet3GPP2Subscription_t::pTrueIMSI`

## 8.873 `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`
- `swi_uint256_t` `ParamPresenceMask`

### 8.873.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
-------------------	--

<i>pUMTSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UMTSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMAInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoIntrafreq</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoInterfreq</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringGSM</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringWCDMA</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pWCDMAInfoLT-ENeighborCell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMAInfoLTENeighborCell</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.873.2 Field Documentation

8.873.2.1 `swi_uint256_t unpack_nas_SLQSNasGetCellLocationInfo_t::ParamPresenceMask`

8.873.2.2 `nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pCDMAInfo`

8.873.2.3 `nas_GERANInfo* unpack_nas_SLQSNasGetCellLocationInfo_t::pGERANInfo`

8.873.2.4 `nas_LTEInfoInterfreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoInterfreq`

8.873.2.5 `nas_LTEInfoIntrafreq* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoIntrafreq`

8.873.2.6 `nas_LTEInfoNeighboringGSM* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringGSM`

8.873.2.7 `nas_LTEInfoNeighboringWCDMA* unpack_nas_SLQSNasGetCellLocationInfo_t::pLTEInfoNeighboringWCDMA`

8.873.2.8 `uint32_t*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSCellID`

8.873.2.9 `nas_UMTSInfo*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pUMTSInfo`

8.873.2.10 `nas_WCDMAInfoLTENeighborCell*` `unpack_nas_SLQSNasGetCellLocationInfo_t::pWCDMAInfoLTENeighborCell`

## 8.874 `unpack_nas_SLQSNasGetCellLocationInfoV2_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`
- `nas_WCDMACellInfoExt` \* `pWCDMACellInfoExt`
- `nas_UMTSExtInfo` \* `pUMTSExtInfo`
- `nas_LteEarfcnInfo` \* `pLteEarfcnInfo`
- `swi_uint256_t` `ParamPresenceMask`

### 8.874.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pUMTSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UMTSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pCDMAInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMAInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pLTEInfo-Intrafreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoIntrafreq</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pLTEInfo-Interfreq</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoInterfreq</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pLTEInfo-NeighboringGSM</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringGSM</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>

<i>pLTEInfo-NeighboringWCDMA</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfoNeighboringWCDMA</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pWCDMAInfoLT-ENeighborCell</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMAInfoLTENeighborCell</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pWCDMACell-InfoExt</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMACellInfoExt</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pUMTSExtInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_UMTSExtInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>
<i>pLteEarfcnInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">na_LteEarfcnInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.874.2 Field Documentation

8.874.2.1 `swi_uint256_t unpack_nas_SLQSNasGetCellLocationInfoV2_t::ParamPresenceMask`

8.874.2.2 `nas_CDMAInfo* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pCDMAInfo`

8.874.2.3 `nas_GERANInfo* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pGERANInfo`

8.874.2.4 `nas_LteEarfcnInfo* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pLteEarfcnInfo`

8.874.2.5 `nas_LTEInfoInterfreq* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pLTEInfoInterfreq`

8.874.2.6 `nas_LTEInfoIntrafreq* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pLTEInfoIntrafreq`

8.874.2.7 `nas_LTEInfoNeighboringGSM* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pLTEInfoNeighboringGSM`

8.874.2.8 `nas_LTEInfoNeighboringWCDMA* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pLTEInfoNeighboringWCDMA`

8.874.2.9 `uint32_t* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pUMTSCellID`

8.874.2.10 `nas_UMTSExtInfo* unpack_nas_SLQSNasGetCellLocationInfoV2_t::pUMTSExtInfo`

8.874.2.11 nas\_UMTSInfo\* unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t::pUMTSInfo

8.874.2.12 nas\_WCDMACellInfoExt\* unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t::pWCDMACellInfoExt

8.874.2.13 nas\_WCDMAInfoLTENeighborCell\* unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t::pWCDMAInfoLTE-NeighborCell

## 8.875 unpack\_nas\_SLQSNASGeteDRXParams\_t Struct Reference

### Data Fields

- uint8\_t \* [pCycleLen](#)
- uint8\_t \* [pPagingTimeWindow](#)
- uint8\_t \* [pEdrxEnable](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.875.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams unpack parameters.

#### Parameters

<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX cycle length</li> <li>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPagingTimeWindow</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX paging time window</li> <li>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX Enable</li> <li>• Values               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.875.2 Field Documentation

8.875.2.1 swi\_uint256\_t unpack\_nas\_SLQSNASGeteDRXParams\_t::ParamPresenceMask

8.875.2.2 uint8\_t\* unpack\_nas\_SLQSNASGeteDRXParams\_t::pCycleLen

8.875.2.3 uint8\_t\* unpack\_nas\_SLQSNASGeteDRXParams\_t::pEdrxEnable

8.875.2.4 uint8\_t\* unpack\_nas\_SLQSNASGeteDRXParams\_t::pPagingTimeWindow



## 8.876 unpack\_nas\_SLQSNASGeteDRXParamsExt\_t Struct Reference

### Data Fields

- uint8\_t \* [pCycleLen](#)
- uint8\_t \* [pPagingTimeWindow](#)
- uint8\_t \* [pEdrxEnable](#)
- uint8\_t \* [pEdrxRAT](#)
- uint32\_t \* [pLteOpMode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.876.1 Detailed Description

This structure contains the SLQSNASGeteDRXParams unpack parameters.

#### Parameters

<i>pCycleLen</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX cycle length</li> <li>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPagingTimeWindow</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX paging time window</li> <li>• Value is per 3GPP TS 24.008 Rel-13 Section 10.5.5.32</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pEdrxEnable</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX Enable</li> <li>• Values               <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pEdrxRAT</i>	[Optional] <ul style="list-style-type: none"> <li>• eDRX Radio Access Technology</li> <li>• Values               <ul style="list-style-type: none"> <li>– NAS_RADIO_IF_NO_SVC (0x00) - None (no service)</li> <li>– NAS_RADIO_IF_CDMA_1X (0x01) - cdma2000 @ 1X</li> <li>– NAS_RADIO_IF_CDMA_1XEVD0 (0x02) - cdma2000 @ HRPD (1xEV-DO)</li> <li>– NAS_RADIO_IF_AMPS (0x03) - AMPS</li> <li>– NAS_RADIO_IF_GSM (0x04) - GSM</li> <li>– NAS_RADIO_IF_UMTS (0x05) - UMTS</li> <li>– NAS_RADIO_IF_WLAN (0x06) - WLAN</li> <li>– NAS_RADIO_IF_GPS (0x07) - GPS</li> <li>– NAS_RADIO_IF_LTE (0x08) - LTE</li> <li>– NAS_RADIO_IF_TDSCDMA (0x09) - TD-SCDMA</li> <li>– NAS_RADIO_IF_LTE_M1 (0x0a) - LTE-M1</li> <li>– NAS_RADIO_IF_LTE_NB1 (0x0b) - LTE-NB1</li> <li>– NAS_RADIO_IF_NO_CHANGE (-1) - No change</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>pLteOpMode</i>	[Optional] <ul style="list-style-type: none"> <li>• LTE Operational Mode</li> <li>• Values             <ul style="list-style-type: none"> <li>– NAS_CIoT_SYS_MODE_NO_SRV (0x00) - No service</li> <li>– NAS_CIoT_SYS_MODE_LTE_WB (0x01) - Camped on LTE wideband</li> <li>– NAS_CIoT_SYS_MODE_LTE_M1 (0x02) - Camped on LTE M1</li> <li>– NAS_CIoT_SYS_MODE_LTE_NB1 (0x03) - Camped on LTE NB1</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.876.2 Field Documentation

8.876.2.1 `swi_uint256_t unpack_nas_SLQSNASGetDRXParamsExt_t::ParamPresenceMask`

8.876.2.2 `uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pCycleLen`

8.876.2.3 `uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pEdrxEnable`

8.876.2.4 `uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pEdrxRAT`

8.876.2.5 `uint32_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pLteOpMode`

8.876.2.6 `uint8_t* unpack_nas_SLQSNASGetDRXParamsExt_t::pPagingTimeWindow`

## 8.877 `unpack_nas_SLQSNASGetForbiddenNetworks_t` Struct Reference

### Data Fields

- [nas\\_ForbiddenNetworks3GPP](#) \* `pForbiddenNetworks3GPP`
- `swi_uint256_t ParamPresenceMask`

### 8.877.1 Detailed Description

This structure contains the SLQSNASGetDRXParams request parameters.

#### Parameters

<i>pForbiddenNetworks3GPP</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_ForbiddenNetworks3GPP</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.877.2 Field Documentation

8.877.2.1 `swi_uint256_t unpack_nas_SLQSNASGetForbiddenNetworks_t::ParamPresenceMask`

8.877.2.2 nas\_ForbiddenNetworks3GPP\* unpack\_nas\_SLQSNASGetForbiddenNetworks\_t::pForbiddenNetworks3GPP

## 8.878 unpack\_nas\_SLQSNasGetHDRColorCode\_t Struct Reference

### Data Fields

- [uint8\\_t \\* pColorCode](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.878.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.878.2 Field Documentation

8.878.2.1 [swi\\_uint256\\_t](#) unpack\_nas\_SLQSNasGetHDRColorCode\_t::ParamPresenceMask

8.878.2.2 [uint8\\_t\\*](#) unpack\_nas\_SLQSNasGetHDRColorCode\_t::pColorCode

## 8.879 unpack\_nas\_SLQSNasGetRFInfo\_t Struct Reference

### Data Fields

- [nas\\_RfBandInfoList](#) rfbandInfoList
- [nas\\_RfDedicatedBandInfo](#) \* pRfDedicatedBandInfo
- [nas\\_RfBandInfoExtFormat](#) \* pRfBandInfoExtFormat
- [nas\\_RfBandwidthInfo](#) \* pRfBandwidthInfo
- [nas\\_LTEOperationMode](#) \* pLTEOperationMode
- [uint16\\_t](#) Tlvresult

### 8.879.1 Detailed Description

This structure contains the SLQSNasGetRFInfo response parameters.

#### Parameters

<i>rfbandInfoList</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RfBandInfoList</a> for more information</li> </ul>
-----------------------	---

<i>pRfDedicated-BandInfo</i>	[Optional] • See <a href="#">nas_RfDedicatedBandInfo</a> for more information
<i>pRfBandInfoExt-Format</i>	[Optional] • See <a href="#">nas_RfBandInfoExtFormat</a> for more information
<i>pRfBandwidth-Info</i>	[Optional] • See <a href="#">nas_RfBandwidthInfo</a> for more information
<i>pLTEOperation-Mode</i>	[Optional] • See <a href="#">nas_LTEOperationMode</a> for more information

## 8.879.2 Field Documentation

8.879.2.1 `nas_LTEOperationMode*` `unpack_nas_SLQSNasGetRFInfo_t::pLTEOperationMode`

8.879.2.2 `nas_RfBandInfoExtFormat*` `unpack_nas_SLQSNasGetRFInfo_t::pRfBandInfoExtFormat`

8.879.2.3 `nas_RfBandwidthInfo*` `unpack_nas_SLQSNasGetRFInfo_t::pRfBandwidthInfo`

8.879.2.4 `nas_RfDedicatedBandInfo*` `unpack_nas_SLQSNasGetRFInfo_t::pRfDedicatedBandInfo`

8.879.2.5 `nas_RfBandInfoList` `unpack_nas_SLQSNasGetRFInfo_t::rfbandInfoList`

8.879.2.6 `uint16_t` `unpack_nas_SLQSNasGetRFInfo_t::Tlvresult`

## 8.880 `unpack_nas_SLQSNasGetSigInfo_t` Struct Reference

### Data Fields

- [cdmaSSInfo](#) `CDMASSInfo`
- [hdrSSInfo](#) `HDRSSInfo`
- `int8_t` `GSMSSInfo`
- [cdmaSSInfo](#) `WCDMASSInfo`
- [lteSSInfo](#) `LTESSInfo`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.880.1 Detailed Description

This structure contains the `SLQSNasGetSigInfo` response parameters.

#### Parameters

<i>CDMASSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cdmaSSInfo</a> for more information</li> <li>• Bit to check in <code>ParamPresenceMask</code> - <b>16</b></li> </ul>
<i>HDRSSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">hdrSSInfo</a> for more information</li> <li>• Bit to check in <code>ParamPresenceMask</code> - <b>17</b></li> </ul>

<i>GSMSSInfo</i>	<ul style="list-style-type: none"> <li>• GSM signal strength is the RSSI in dBm (signed value).</li> <li>• A value of -125 dBm or lower is used to indicate No Signal.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>WCDMASSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cdmaSSInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>LTESSInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteSSInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.880.2 Field Documentation

8.880.2.1 `cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::CDMASSInfo`

8.880.2.2 `int8_t unpack_nas_SLQSNasGetSigInfo_t::GSMSSInfo`

8.880.2.3 `hdrSSInfo unpack_nas_SLQSNasGetSigInfo_t::HDRSSInfo`

8.880.2.4 `lteSSInfo unpack_nas_SLQSNasGetSigInfo_t::LTESSInfo`

8.880.2.5 `swi_uint256_t unpack_nas_SLQSNasGetSigInfo_t::ParamPresenceMask`

8.880.2.6 `cdmaSSInfo unpack_nas_SLQSNasGetSigInfo_t::WCDMASSInfo`

## 8.881 unpack\_nas\_SLQSNasGetTxRxInfo\_t Struct Reference

### Data Fields

- [nas\\_rxInfo](#) \* [pRXChain0Info](#)
- [nas\\_rxInfo](#) \* [pRXChain1Info](#)
- [nas\\_txInfo](#) \* [pTXInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.881.1 Detailed Description

Structure for TxRx Info unpack

#### Parameters

<i>pRXChain0Info</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_rxInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
----------------------	--

<i>pRXChain1Info</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_rxInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pTXInfo</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_txInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.881.2 Field Documentation

8.881.2.1 `swi_uint256_t unpack_nas_SLQSNasGetTxRxInfo_t::ParamPresenceMask`

8.881.2.2 `nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain0Info`

8.881.2.3 `nas_rxInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pRXChain1Info`

8.881.2.4 `nas_txInfo* unpack_nas_SLQSNasGetTxRxInfo_t::pTXInfo`

## 8.882 unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t Struct Reference

### Data Fields

- `uint8_t radiolf`
- `LITE_TYPE_OF_SERVICE_DOMAIN serviceDomain`
- `uint8_t rejectCause`
- `nas_PlmnID * pPlmnId`
- `nas_CsgId * pCsgId`
- `nas_LteOpMode * pLteOpMode`
- `swi_uint256_t ParamPresenceMask`

### 8.882.1 Detailed Description

Structure for storing the network reject indication parameters.

#### Parameters

<i>radiolf</i>	<ul style="list-style-type: none"> <li>• Radio interface from which to get the information. Values:               <ul style="list-style-type: none"> <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 - TDSCDMA</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
----------------	--

<i>serviceDomain</i>	<ul style="list-style-type: none"><li>• Type of service domain in which the registration is rejected. Values:<ul style="list-style-type: none"><li>– 0x00 - LITE_SYS_SRV_DOMAIN_NO_SRV - No service</li><li>– 0x01 - LITE_SYS_SRV_DOMAIN_CS_ONLY - Circuit-switched only</li><li>– 0x02 - LITE_SYS_SRV_DOMAIN_PS_ONLY - Packet-switched only</li><li>– 0x03 - LITE_SYS_SRV_DOMAIN_CS_PS - Circuit-switched and packet-switched</li><li>– 0x04 - LITE_SYS_SRV_DOMAIN_CAMPED - Camped</li></ul></li><li>• Bit to check in ParamPresenceMask - 2</li></ul>
----------------------	---

<i>rejectCause</i>	<ul style="list-style-type: none"> <li>• Reject cause 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>– 7 - EPS services not allowed</li> <li>– 8 - EPS services and non-EPS services not allowed</li> <li>– 9 - UE identity cannot be derived by the network</li> <li>– 10 - Implicitly detached</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>– 14 - EPS services not allowed in this PLMN</li> <li>– 15 - No Suitable Cells In Location Area</li> <li>– 16 - MSC temporarily not reachable</li> <li>– 17 - Network failure</li> <li>– 18 - CS domain not available</li> <li>– 19 - ESM failure</li> <li>– 20 - MAC failure</li> <li>– 21 - Synch failure</li> <li>– 22 - Congestion</li> <li>– 23 - GSM authentication unacceptable</li> <li>– 24 - Security mode rejected, unspecified</li> <li>– 25 - Not authorized for this CSG</li> <li>– 26 - Non-EPS authentication unacceptable</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> <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> </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> <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
--------------------	---



<i>pPlmnId</i>	(optional) <ul style="list-style-type: none"> <li>See <a href="#">nas_PlmnID</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pCsgId</i>	(optional) <ul style="list-style-type: none"> <li>See <a href="#">nas_CsgId</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pLteOpMode</i>	(optional) <ul style="list-style-type: none"> <li>See <a href="#">nas_LteOpMode</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.882.2 Field Documentation

8.882.2.1 `swi_uint256_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::ParamPresenceMask`

8.882.2.2 `nas_CsgId* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pCsgId`

8.882.2.3 `nas_LteOpMode* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pLteOpMode`

8.882.2.4 `nas_PlmnID* unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::pPlmnId`

8.882.2.5 `uint8_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::radioIcf`

8.882.2.6 `uint8_t unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::rejectCause`

8.882.2.7 `LITE_TYPE_OF_SERVICE_DOMAIN unpack_nas_SLQSNasNetworkRejectCallback_Ind_t::serviceDomain`

## 8.883 unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t Struct Reference

### Data Fields

- [nas\\_UniversalTime](#) `universalTime`
- `uint8_t * pTimeZone`
- `uint8_t * pDayltSavAdj`
- `uint8_t * pRadioInterface`
- `swi_uint256_t ParamPresenceMask`

### 8.883.1 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> <li>Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.883.2 Field Documentation

8.883.2.1 `swi_uint256_t unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::ParamPresenceMask`

8.883.2.2 `uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pDayltSavAdj`

8.883.2.3 `uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pRadioInterface`

8.883.2.4 `uint8_t* unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::pTimeZone`

8.883.2.5 `nas_UniversalTime unpack_nas_SLQSNasNetworkTimeCallBack_ind_t::universalTime`

## 8.884 `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t` Struct Reference

### Data Fields

- [nas\\_RFBandInfoTlv](#) `rfBandInfo`
- [nas\\_RFDedicatedBandInfoTlv](#) `rfDedicatedBandInfo`
- [nas\\_RFBandInfoExtTlv](#) `rfBandInfoExt`
- [nas\\_RFBandwidthInfoTlv](#) `rfBandwidthInfo`
- [nas\\_LTEOperationalModeTlv](#) `lteOperationalMode`
- `swi_uint256_t` `ParamPresenceMask`

### 8.884.1 Detailed Description

Contain the parameters passed for unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind by the device.

#### Parameters

<i>rfBandInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RFBandInfoTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>rfDedicated-BandInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RFDedicatedBandInfoTlv</a> for more information.</li> </ul>

- Bit to check in ParamPresenceMask - **16**

#### Parameters

<i>rfBandInfoExt</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RFBandInfoExtTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>rfBandwidthInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RFBandwidthInfoTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>lteOperational-Mode</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEOperationalModeTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

#### Note

None

### 8.884.2 Field Documentation

8.884.2.1 `nas_LTEOperationalModeTlv` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::lteOperationalMode`

8.884.2.2 `swi_uint256_t` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::ParamPresenceMask`

8.884.2.3 `nas_RFBandInfoTlv` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::rfBandInfo`

8.884.2.4 `nas_RFBandInfoExtTlv` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::rfBandInfoExt`

8.884.2.5 `nas_RFBandwidthInfoTlv` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::rfBandwidthInfo`

8.884.2.6 `nas_RFDedicatedBandInfoTlv` `unpack_nas_SLQSNasRFBandInfoCallback_Ind_t::rfDedicatedBandInfo`

## 8.885 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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.885.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHDRSigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">hdrSSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pWCDMASigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">cdmaSSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pLTESigInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">lteSSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pTDSCDMASig-InfoExt</i>	<ul style="list-style-type: none"> <li>• See <a href="#">tdscdmaSigInfoExt</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.885.2 Field Documentation

8.885.2.1 [swi\\_uint256\\_t](#) [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::ParamPresenceMask](#)

8.885.2.2 [cdmaSSInfo\\*](#) [unpack\\_nas\\_SLQSNasSigInfoCallback\\_ind\\_t::pCDMASigInfo](#)

8.885.2.3 `int8_t*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pGMSSigInfo`

8.885.2.4 `hdrSSInfo*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pHDRSigInfo`

8.885.2.5 `lteSSInfo*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pLTESigInfo`

8.885.2.6 `int8_t*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pRscp`

8.885.2.7 `tdscdmaSigInfoExt*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pTDSCDMASigInfoExt`

8.885.2.8 `cdmaSSInfo*` `unpack_nas_SLQSNasSigInfoCallback_ind_t::pWCDMASigInfo`

## 8.886 unpack\_nas\_SLQSNASSwiGetChannelLock\_t Struct Reference

### Data Fields

- [nas\\_wcdmaUARFCN](#) \* [pWcdmaUARFCN](#)
- [nas\\_lteEARFCN](#) \* [pLteEARFCN](#)
- [nas\\_ltePCI](#) \* [pLtePCI](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.886.1 Detailed Description

Structure for Get Channel Lock unpack.

#### Parameters

<i>pWcdmaUARFCN</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_wcdmaUARFCN</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pLteEARFCN</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_lteEARFCN</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pLtePCI</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">nas_ltePCI</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.886.2 Field Documentation

8.886.2.1 `swi_uint256_t` `unpack_nas_SLQSNASSwiGetChannelLock_t::ParamPresenceMask`

8.886.2.2 `nas_lteEARFCN*` `unpack_nas_SLQSNASSwiGetChannelLock_t::pLteEARFCN`

8.886.2.3 `nas_ltePCI*` `unpack_nas_SLQSNASSwiGetChannelLock_t::pLtePCI`

8.886.2.4 `nas_wcdmaUARFCN*` `unpack_nas_SLQSNASSwiGetChannelLock_t::pWcdmaUARFCN`

## 8.887 unpack\_nas\_SLQSNasSwiModemStatus\_t Struct Reference

### Data Fields

- [nas\\_CommInfo](#) [commonInfo](#)
- [nas\\_LTEInfo](#) \* [pLTEInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.887.1 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="#">nas_CommInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pLTEInfo</i>	(optional) <ul style="list-style-type: none"> <li>• See <a href="#">nas_LTEInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.887.2 Field Documentation

8.887.2.1 [nas\\_CommInfo](#) [unpack\\_nas\\_SLQSNasSwiModemStatus\\_t::commonInfo](#)

8.887.2.2 [swi\\_uint256\\_t](#) [unpack\\_nas\\_SLQSNasSwiModemStatus\\_t::ParamPresenceMask](#)

8.887.2.3 [nas\\_LTEInfo\\*](#) [unpack\\_nas\\_SLQSNasSwiModemStatus\\_t::pLTEInfo](#)

## 8.888 unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t Struct Reference

### Data Fields

- [NASQmiCbkJasSwiOTAMessageInd](#) [Info](#)
- [uint16\\_t](#) [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.888.1 Detailed Description

This structure contains unpack OTA message indication parameters.

#### 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="#">NASQmiCbkJasSwiOTAMessageInd</a> for more details</li> </ul> </li> </ul>
-------------	---

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.888.2 Field Documentation

8.888.2.1 NASQmiCbkNasSwiOTAMessageInd unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t::Info

8.888.2.2 swi\_uint256\_t unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t::ParamPresenceMask

8.888.2.3 uint16\_t unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t::Tlvresult

## 8.889 unpack\_nas\_SLQSNasTimerCallback\_ind\_t Struct Reference

### Data Fields

- char [t3396\\_apn](#) [101]
- uint8\_t [t3396\\_plmn\\_id](#) [3]
- uint32\_t [t3396\\_val](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.889.1 Detailed Description

Structure for Network Timer indication parameters.

#### Parameters

<i>t3396_apn</i>	<ul style="list-style-type: none"> <li>• apn</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>t3396_plmn_id</i>	<ul style="list-style-type: none"> <li>• plmn id</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>t3396_val</i>	<ul style="list-style-type: none"> <li>• timer value</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.889.2 Field Documentation

8.889.2.1 swi\_uint256\_t unpack\_nas\_SLQSNasTimerCallback\_ind\_t::ParamPresenceMask

8.889.2.2 char unpack\_nas\_SLQSNasTimerCallback\_ind\_t::t3396\_apn[101]

8.889.2.3 uint8\_t unpack\_nas\_SLQSNasTimerCallback\_ind\_t::t3396\_plmn\_id[3]

8.889.2.4 uint32\_t unpack\_nas\_SLQSNasTimerCallback\_ind\_t::t3396\_val

## 8.890 unpack\_nas\_SLQSPerformNetworkScanV2\_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](#)
- [nas\\_QmisNasSlqsNasPCInfo](#) \* [pPCInfo](#)
- [nas\\_lteOpModeTlv](#) \* [pLteOpModeTlv](#)
- [nas\\_networkNameSrcTlv](#) \* [pNetworkNameSrcTlv](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.890.1 Detailed Description

Contain the network scan information.

#### Parameters

<i>p3GppNetwork-InstanceSize</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>p3GppNetwork-InfoInstances</i>	<ul style="list-style-type: none"> <li>• Network info instance array <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmiNas3GppNetworkInfo</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pRATInstance-Size</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pRATInstance</i>	<ul style="list-style-type: none"> <li>• RAT info instance array <ul style="list-style-type: none"> <li>– See <a href="#">nas_QmiNas3GppNetworkRAT</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pPCSInstance-Size</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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>



<i>pPCSInstance</i>	<ul style="list-style-type: none"> <li>PCS Digit info instance array <ul style="list-style-type: none"> <li>See <a href="#">nas_QmisNasPcsDigit</a> for more information</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pScanResult</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> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pPCIInfo</i>	<ul style="list-style-type: none"> <li>PCI Information <ul style="list-style-type: none"> <li>See <a href="#">nas_QmisNasSlqsNasPCIInfo</a> for more information</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pLteOpMode-Tlv[OUT]</i>	<ul style="list-style-type: none"> <li>LTE Operational Mode.</li> <li>See <a href="#">nas_LteOpModeTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pNetworkName-SrcTlv[OUT]</i>	<ul style="list-style-type: none"> <li>Network Name Source.</li> <li>See <a href="#">nas_networkNameSrcTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.890.2 Field Documentation

8.890.2.1 `nas_QmiNas3GppNetworkInfo*` `unpack_nas_SLQSPerformNetworkScanV2_t::p3GppNetworkInfoInstances`

8.890.2.2 `uint8_t*` `unpack_nas_SLQSPerformNetworkScanV2_t::p3GppNetworkInstanceSize`

8.890.2.3 `swi_uint256_t` `unpack_nas_SLQSPerformNetworkScanV2_t::ParamPresenceMask`

8.890.2.4 `nas_LteOpModeTlv*` `unpack_nas_SLQSPerformNetworkScanV2_t::pLteOpModeTlv`

8.890.2.5 `nas_networkNameSrcTlv*` `unpack_nas_SLQSPerformNetworkScanV2_t::pNetworkNameSrcTlv`

8.890.2.6 `nas_QmisNasSlqsNasPCIInfo*` `unpack_nas_SLQSPerformNetworkScanV2_t::pPCIInfo`

8.890.2.7 `nas_QmisNasPcsDigit*` `unpack_nas_SLQSPerformNetworkScanV2_t::pPCSInstance`

8.890.2.8 `uint8_t*` `unpack_nas_SLQSPerformNetworkScanV2_t::pPCSInstanceSize`

8.890.2.9 `nas_QmiNas3GppNetworkRAT*` `unpack_nas_SLQSPerformNetworkScanV2_t::pRATInstance`

8.890.2.10 `uint8_t*` `unpack_nas_SLQSPerformNetworkScanV2_t::pRATInstanceSize`

8.890.2.11 uint32\_t\* unpack\_nas\_SLQSPerformNetworkScanV2\_t::pScanResult

## 8.891 unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t Struct Reference

### Data Fields

- [NASQmiCbkJasSystemSelPrefInd Info](#)
- uint16\_t Tlvresult
- swi\_uint256\_t ParamPresenceMask

### 8.891.1 Detailed Description

This structure contains unpack set system selection preference callback indication parameters.

#### 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="#">NASQmiCbkJasSystemSelPrefInd</a> for more details</li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.891.2 Field Documentation

8.891.2.1 NASQmiCbkJasSystemSelPrefInd unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t::Info

8.891.2.2 swi\_uint256\_t unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t::ParamPresenceMask

8.891.2.3 uint16\_t unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t::Tlvresult

## 8.892 unpack\_nas\_SLQSSwiGetHDRPersonality\_t Struct Reference

### Data Fields

- uint16\_t \* pCurrentPersonality
- uint8\_t \* pPersonalityListLength
- nas\_protocolSubtypeElement \* pProtocolSubtypeElement
- swi\_uint256\_t ParamPresenceMask

### 8.892.1 Detailed Description

Structure for HDRPersonality unpack.

## Parameters

<i>pCurrent-Personality</i> [Out]	<ul style="list-style-type: none"> <li>Current active personality index.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPersonalityList-Length</i> [In/Out]	<ul style="list-style-type: none"> <li>Number of Personality Protocol Subtype contains in this response.</li> <li>maximum input value is 3</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pProtocol-Subtype-Element</i> [Out]	<ul style="list-style-type: none"> <li>See <a href="#">nas_protocolSubtypeElement</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.892.2 Field Documentation

8.892.2.1 swi\_uint256\_t unpack\_nas\_SLQSSwiGetHDRPersonality\_t::ParamPresenceMask

8.892.2.2 uint16\_t\* unpack\_nas\_SLQSSwiGetHDRPersonality\_t::pCurrentPersonality

8.892.2.3 uint8\_t\* unpack\_nas\_SLQSSwiGetHDRPersonality\_t::pPersonalityListLength

8.892.2.4 nas\_protocolSubtypeElement\* unpack\_nas\_SLQSSwiGetHDRPersonality\_t::pProtocolSubtypeElement

## 8.893 unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t Struct Reference

## Data Fields

- uint16\_t \* [pCurrentPrsnlty](#)
- uint8\_t \* [pPersonalityListLength](#)
- [nas\\_protocolSubtypeElement](#) \* [pProtoSubTypElmnt](#)
- uint64\_t \* [pAppSubType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.893.1 Detailed Description

Structure for HDRProtSubtype unpack.

## Parameters

<i>pCurrent-Personality</i>	<ul style="list-style-type: none"> <li>Current active personality index.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pPersonalityList-Length</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> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>pProtocolSubtypeElement</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_protocolSubtypeElement</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAppSubType</i>	<ul style="list-style-type: none"> <li>• Stream application subtype</li> <li>• Application subtype for each stream</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.893.2 Field Documentation

8.893.2.1 `uint64_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pAppSubType`

8.893.2.2 `swi_uint256_t unpack_nas_SLQSSwiGetHDRProtSubtype_t::ParamPresenceMask`

8.893.2.3 `uint16_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pCurrentPrsnlty`

8.893.2.4 `uint8_t* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pPersonalityListLength`

8.893.2.5 `nas_protocolSubtypeElement* unpack_nas_SLQSSwiGetHDRProtSubtype_t::pProtoSubTypeElmnt`

## 8.894 `unpack_nas_SLQSSwiGetHRPDStats_t` Struct Reference

### Data Fields

- [nas\\_DRCParams](#) \* `pDRCParams`
- `uint8_t` \* `pUATI`
- [nas\\_PilotSetData](#) \* `pPilotSetData`
- `swi_uint256_t` `ParamPresenceMask`

### 8.894.1 Detailed Description

Structure for HRPD Stats unpack.

#### Parameters

<i>pDRCParams</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_DRCParams</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pUATI</i>	<ul style="list-style-type: none"> <li>• A 128-bit address that includes the access terminal identifier and subnet ID</li> <li>• Size must be 16 bytes</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pPilotSetData</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_PilotSetData</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.894.2 Field Documentation

8.894.2.1 `swi_uint256_t` `unpack_nas_SLQSSwiGetHRPDStats_t::ParamPresenceMask`

8.894.2.2 `nas_DRCParams*` `unpack_nas_SLQSSwiGetHRPDStats_t::pDRCParams`

8.894.2.3 `nas_PilotSetData*` `unpack_nas_SLQSSwiGetHRPDStats_t::pPilotSetData`

8.894.2.4 `uint8_t*` `unpack_nas_SLQSSwiGetHRPDStats_t::pUATI`

## 8.895 unpack\_nas\_SLQSSwiGetLteCQI\_t Struct Reference

### Data Fields

- `uint8_t` [ValidityCW0](#)
- `uint8_t` [CQIValueCW0](#)
- `uint8_t` [ValidityCW1](#)
- `uint8_t` [CQIValueCW1](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.895.1 Detailed Description

This structure contains paramaters unpack fetch CQI parameters for LTE data session.

#### 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> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.895.2 Field Documentation

8.895.2.1 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW0`

8.895.2.2 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::CQIValueCW1`

8.895.2.3 `swi_uint256_t unpack_nas_SLQSSwiGetLteCQI_t::ParamPresenceMask`

8.895.2.4 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW0`

8.895.2.5 `uint8_t unpack_nas_SLQSSwiGetLteCQI_t::ValidityCW1`

## 8.896 `unpack_nas_SLQSSwiGetLteSccRxInfo_t` Struct Reference

### Data Fields

- `nas_SccRxInfo * pSccRxInfo`
- `swi_uint256_t ParamPresenceMask`

### 8.896.1 Detailed Description

This structure contains unpack get LTE Secondary carrier Rx signal level information parameters.

#### Parameters

<i>pSccRxInfo</i>	Secondary carrier Rx signal level info <ul style="list-style-type: none"> <li>See <a href="#">nas_SccRxInfo</a> <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.896.2 Field Documentation

8.896.2.1 `swi_uint256_t unpack_nas_SLQSSwiGetLteSccRxInfo_t::ParamPresenceMask`

8.896.2.2 `nas_SccRxInfo* unpack_nas_SLQSSwiGetLteSccRxInfo_t::pSccRxInfo`

## 8.897 `unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t` Struct Reference

### Data Fields

- `nas_HDRPersonality_Ind_Data * pHDRPersInd`
- `swi_uint256_t ParamPresenceMask`

### 8.897.1 Detailed Description

Structure for HDR Personality indication unpack.

#### Parameters

<i>pHDRPersInd</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_HDRPersonality_Ind_Data</a> for more information.</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.897.2 Field Documentation

8.897.2.1 `swi_uint256_t unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::ParamPresenceMask`

8.897.2.2 `nas_HDRPersonality_Ind_Data* unpack_nas_SLQSSwiHDRPersonalityCallback_Ind_t::pHDRPersInd`

## 8.898 unpack\_nas\_SLQSSwiNetworkDebug\_t Struct Reference

### Data Fields

- `uint8_t * pObjectVer`
- `nas_NetworkStat1x * pNetworkStat1x`
- `nas_NetworkStatEVDO * pNetworkStatEVDO`
- `nas_DeviceConfigDetail * pDeviceConfigDetail`
- `nas_DataStatusDetail * pDataStatusDetail`
- `swi_uint256_t ParamPresenceMask`

### 8.898.1 Detailed Description

Structure for SwiNetworkDebug unpack.

#### 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pNetworkStat1x</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_NetworkStat1x</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNetworkStatEVDO</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_NetworkStatEVDO</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pDeviceConfigDetail</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_DeviceConfigDetail</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>pDataStatus- Detail</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_DataStatusDetail</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.898.2 Field Documentation

8.898.2.1 `swi_uint256_t` `unpack_nas_SLQSSwiNetworkDebug_t::ParamPresenceMask`

8.898.2.2 `nas_DataStatusDetail*` `unpack_nas_SLQSSwiNetworkDebug_t::pDataStatusDetail`

8.898.2.3 `nas_DeviceConfigDetail*` `unpack_nas_SLQSSwiNetworkDebug_t::pDeviceConfigDetail`

8.898.2.4 `nas_NetworkStat1x*` `unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStat1x`

8.898.2.5 `nas_NetworkStatEVDO*` `unpack_nas_SLQSSwiNetworkDebug_t::pNetworkStatEVDO`

8.898.2.6 `uint8_t*` `unpack_nas_SLQSSwiNetworkDebug_t::pObjectVer`

## 8.899 unpack\_nas\_SLQSSwiRandIndicatorCallback\_Ind\_t Struct Reference

### Data Fields

- [nas\\_RankIndicatorTlv](#) `rankIndicatorTlv`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.899.1 Detailed Description

Structure for Rank Indicator indication unpack.

#### Parameters

<i>rankIndicatorTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_RankIndicatorTlv</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.899.2 Field Documentation

8.899.2.1 `swi_uint256_t` `unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::ParamPresenceMask`

8.899.2.2 `nas_RankIndicatorTlv` `unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t::rankIndicatorTlv`

## 8.900 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](#) \* [pAddGSMSysInfo](#)
- [nas\\_AddSysInfo](#) \* [pAddWCDMASysInfo](#)
- [uint16\\_t](#) \* [pAddLTESysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pGSMCallBarringSysInfo](#)
- [nas\\_CallBarringSysInfo](#) \* [pWCDMACallBarringSysInfo](#)
- [uint8\\_t](#) \* [pLTEVoiceSupportSysInfo](#)
- [uint8\\_t](#) \* [pGSMCipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pWCDMACipherDomainSysInfo](#)
- [uint8\\_t](#) \* [pSysInfoNoChange](#)
- [nas\\_LteCiotOpModeTlv](#) \* [pLteCiotOpModeTlv](#)
- [nas\\_NR5GSerStatTlv](#) \* [pNR5GSerStatTlv](#)
- [nas\\_NR5GSystemInfoTlv](#) \* [pNR5GSystemInfoTlv](#)
- [nas\\_NR5GCellStatusTlv](#) \* [pNR5GCellStatus](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.900.1 Detailed Description

This structure contains unpack system information callback indication parameters.

## Parameters

<i>pCDMASrv- StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pHDRSrvStatus- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_SrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pGSMSrvStatus- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pWCDMASrv- StatusInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></li> </ul>
<i>pLTESrvStatus- Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSrvStatusInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>20</b></li> </ul>

<i>pCDMASysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pHDRSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_HDRSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pGSMSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_GSMSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pWCDMASys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_WCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pLTESysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_LTESysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pAddCDMASys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddCDMASysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pAddGSMSys-Info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pAddWCDMA-SysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_AddSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pGSMCall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pWCDMACall-BarringSysInfo</i>	<ul style="list-style-type: none"> <li>• See <a href="#">nas_CallBarringSysInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></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> <li>Bit to check in ParamPresenceMask - <b>33</b></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> <li>Bit to check in ParamPresenceMask - <b>34</b></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> <li>Bit to check in ParamPresenceMask - <b>35</b></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> <li>Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>pLteCiotOp-ModeTlv</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_LteCiotOpModeTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>75</b></li> </ul>
<i>pNR5GSerStat-Tlv</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_NR5GSerStatTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>76</b></li> </ul>
<i>pNR5GSystem-InfoTlv</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_NR5GSystemInfoTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>77</b></li> </ul>
<i>pNR5GCell-StatusTlv</i>	<ul style="list-style-type: none"> <li>See <a href="#">nas_NR5GCellStatusTlv</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>78</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.900.2 Field Documentation

- 8.900.2.1 **nas\_AddCDMASysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddCDMASysInfo**
- 8.900.2.2 **nas\_AddSysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddGSM SysInfo**
- 8.900.2.3 **uint16\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddHDR SysInfo**
- 8.900.2.4 **uint16\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddLTE SysInfo**
- 8.900.2.5 **nas\_AddSysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pAddWCDMA SysInfo**
- 8.900.2.6 **swi\_uint256\_t** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::ParamPresenceMask**
- 8.900.2.7 **nas\_SrvStatusInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pCDMA SrvStatusInfo**
- 8.900.2.8 **nas\_CDMA SysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pCDMA SysInfo**
- 8.900.2.9 **nas\_CallBarringSysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSM CallBarringSysInfo**
- 8.900.2.10 **uint8\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSM CipherDomainSysInfo**
- 8.900.2.11 **nas\_GSM SrvStatusInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSM SrvStatusInfo**
- 8.900.2.12 **nas\_GSM SysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pGSM SysInfo**
- 8.900.2.13 **nas\_SrvStatusInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pHDR SrvStatusInfo**
- 8.900.2.14 **nas\_HDR SysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pHDR SysInfo**
- 8.900.2.15 **nas\_LteCiotOpModeTlv\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLteCiotOpModeTlv**
- 8.900.2.16 **nas\_GSM SrvStatusInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTE SrvStatusInfo**
- 8.900.2.17 **nas\_LTE SysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTE SysInfo**
- 8.900.2.18 **uint8\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pLTE VoiceSupportSysInfo**
- 8.900.2.19 **nas\_NR5GCellStatusTlv\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pNR5GCellStatus**
- 8.900.2.20 **nas\_NR5GSerStatTlv\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pNR5GSerStatTlv**
- 8.900.2.21 **nas\_NR5GSystemInfoTlv\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pNR5GSystemInfoTlv**
- 8.900.2.22 **uint8\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pSysInfoNoChange**
- 8.900.2.23 **nas\_CallBarringSysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMA CallBarringSysInfo**
- 8.900.2.24 **uint8\_t\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMA CipherDomainSysInfo**
- 8.900.2.25 **nas\_GSM SrvStatusInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMA SrvStatusInfo**
- 8.900.2.26 **nas\_WCDMA SysInfo\*** **unpack\_nas\_SLQSSysInfoCallback\_ind\_t::pWCDMA SysInfo**

## 8.901 **unpack\_omaDmConfigTlv\_t Struct Reference**

## Data Fields

- uint8\_t [state](#)
- uint8\_t [userInputReq](#)
- uint16\_t [userInputTimeout](#)
- uint16\_t [alertmsglength](#)
- uint8\_t [alertmsg](#) [256]

### 8.901.1 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>userInputReq</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>userInput-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>

### 8.901.2 Field Documentation

8.901.2.1 uint8\_t unpack\_omaDmConfigTlv\_t::alertmsg[256]

8.901.2.2 uint16\_t unpack\_omaDmConfigTlv\_t::alertmsglength

8.901.2.3 uint8\_t unpack\_omaDmConfigTlv\_t::state

8.901.2.4 uint8\_t unpack\_omaDmConfigTlv\_t::userInputReq

8.901.2.5 uint16\_t unpack\_omaDmConfigTlv\_t::userInputTimeout

## 8.902 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](#)

### 8.902.1 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>userInputReq</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>userInput-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>fwloadsize</i>	<ul style="list-style-type: none"> <li>• The size (in bytes) of the firmware update package</li> </ul>
<i>fwloadComplete</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-CompleteStatus</i>	<ul style="list-style-type: none"> <li>• This field should be looked at only when the OMADM session is complete.</li> <li>• See <a href="#">qaGobiApiTableSwiOMADMUpdateCompleteStatus.h</a> for update complete status.</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>

## 8.902.2 Field Documentation

- 8.902.2.1 `uint8_t unpack_omaDmFotaTlv_t::description[256]`
- 8.902.2.2 `uint16_t unpack_omaDmFotaTlv_t::descriptionlength`
- 8.902.2.3 `uint32_t unpack_omaDmFotaTlv_t::fwdloadsize`
- 8.902.2.4 `uint32_t unpack_omaDmFotaTlv_t::fwloadComplete`
- 8.902.2.5 `uint16_t unpack_omaDmFotaTlv_t::namelength`
- 8.902.2.6 `uint8_t unpack_omaDmFotaTlv_t::package_name[256]`
- 8.902.2.7 `uint8_t unpack_omaDmFotaTlv_t::sessionType`
- 8.902.2.8 `uint8_t unpack_omaDmFotaTlv_t::severity`
- 8.902.2.9 `uint8_t unpack_omaDmFotaTlv_t::state`
- 8.902.2.10 `uint16_t unpack_omaDmFotaTlv_t::updateCompleteStatus`
- 8.902.2.11 `uint8_t unpack_omaDmFotaTlv_t::userInputReq`
- 8.902.2.12 `uint16_t unpack_omaDmFotaTlv_t::userInputTimeout`
- 8.902.2.13 `uint8_t unpack_omaDmFotaTlv_t::version[256]`
- 8.902.2.14 `uint16_t unpack_omaDmFotaTlv_t::versionlength`

## 8.903 unpack\_omaDmNotificationsTlv\_t Struct Reference

### Data Fields

- `uint8_t` [notification](#)
- `uint16_t` [sessionStatus](#)

### 8.903.1 Detailed Description

This structure will hold the SwiOmaDmConfig session notification parameters information.

#### Parameters

<i>notification</i>	<ul style="list-style-type: none"> <li>• 0x01 - GPS settings change</li> <li>• 0x02 - Device reset</li> <li>• 0x03 - Device factory reset</li> <li>• 0x04 - CI-DC Session start</li> <li>• 0x05 - CI-DC Session end</li> <li>• 0x06 - CI-PRL Session start</li> <li>• 0x07 - CI-PRL Session end</li> <li>• 0x08 - CI-FUMO Session start</li> <li>• 0x09 - CI-FUMO session end</li> <li>• 0x0A - HFA-DC Session start</li> <li>• 0x0B - HFA-DC Session end</li> <li>• 0x0C - HFA-PRL Session start</li> <li>• 0x0D - HFA-PRL Session end</li> <li>• 0x0E - HFA-FUMO Session start</li> <li>• 0x0F - HFA-FUMO session end</li> <li>• 0x10 - NI Session start</li> <li>• 0x11 - NI session end</li> </ul>
<i>sessionStatus</i>	<ul style="list-style-type: none"> <li>• This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications</li> <li>• See <a href="#">qaGobiApiTableSwiOMADMSessionStatus.h</a> for session completion code.</li> </ul>

### 8.903.2 Field Documentation

8.903.2.1 `uint8_t unpack_omaDmNotificationsTlv_t::notification`

8.903.2.2 `uint16_t unpack_omaDmNotificationsTlv_t::sessionStatus`

## 8.904 unpack\_pds\_GetPDSDefaults\_t Struct Reference

### Data Fields

- `uint32_t * pOperation`
- `uint8_t * pTimeout`
- `uint32_t * pInterval`
- `uint32_t * pAccuracy`
- `swi_uint256_t ParamPresenceMask`

### 8.904.1 Detailed Description

Structure contain session configuration parameters.



## Parameters

<i>pOperation</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> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pTimeout</i>	<ul style="list-style-type: none"> <li>Maximum amount of time (seconds) to work on each fix, maximum is 255</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pInterval</i>	<ul style="list-style-type: none"> <li>Interval (seconds) between fix requests</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pAccuracy</i>	<ul style="list-style-type: none"> <li>Preferred accuracy threshold (meters)</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.904.2 Field Documentation

8.904.2.1 uint32\_t\* unpack\_pds\_GetPDSDDefaults\_t::pAccuracy

8.904.2.2 swi\_uint256\_t unpack\_pds\_GetPDSDDefaults\_t::ParamPresenceMask

8.904.2.3 uint32\_t\* unpack\_pds\_GetPDSDDefaults\_t::pInterval

8.904.2.4 uint32\_t\* unpack\_pds\_GetPDSDDefaults\_t::pOperation

8.904.2.5 uint8\_t\* unpack\_pds\_GetPDSDDefaults\_t::pTimeout

## 8.905 unpack\_pds\_GetPDSSState\_t Struct Reference

## Data Fields

- uint32\_t \* [pEnabledStatus](#)
- uint32\_t \* [pTrackingStatus](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.905.1 Detailed Description

Structure contain parameters of current PDS state.

## Parameters

<i>pEnabledStatus</i>	<ul style="list-style-type: none"> <li>Current PDS state <ul style="list-style-type: none"> <li>0 - disable</li> <li>1 - enable</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pTrackingStatus</i>	<ul style="list-style-type: none"> <li>Current PDS tracking session state</li> <li>Values: <ul style="list-style-type: none"> <li>0x00 - Unknown</li> <li>0x01 - Inactive</li> <li>0x02 - Active</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.905.2 Field Documentation

8.905.2.1 `swi_uint256_t unpack_pds_GetPDSSState_t::ParamPresenceMask`8.905.2.2 `uint32_t* unpack_pds_GetPDSSState_t::pEnabledStatus`8.905.2.3 `uint32_t* unpack_pds_GetPDSSState_t::pTrackingStatus`8.906 `unpack_pds_GetPortAutomaticTracking_t` Struct Reference

## Data Fields

- `uint32_t * pbAuto`
- `swi_uint256_t ParamPresenceMask`

## 8.906.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

## Parameters

<i>pbAuto</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.906.2 Field Documentation

8.906.2.1 `swi_uint256_t` unpack\_pds\_GetPortAutomaticTracking\_t::ParamPresenceMask

8.906.2.2 `uint32_t*` unpack\_pds\_GetPortAutomaticTracking\_t::pbAuto

## 8.907 unpack\_pds\_GetServiceAutomaticTracking\_t Struct Reference

### Data Fields

- `uint32_t *` [pbAuto](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.907.1 Detailed Description

Structure contain the parameter for automatic tracking state.

#### Parameters

<i>pbAuto</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><li>• Bit to check in ParamPresenceMask - <b>1</b></li></ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>

### 8.907.2 Field Documentation

8.907.2.1 `swi_uint256_t` unpack\_pds\_GetServiceAutomaticTracking\_t::ParamPresenceMask

8.907.2.2 `uint32_t*` unpack\_pds\_GetServiceAutomaticTracking\_t::pbAuto

## 8.908 unpack\_pds\_GetXTRAAutomaticDownload\_t Struct Reference

### Data Fields

- `uint32_t *` [pbEnabled](#)
- `uint16_t *` [pInterval](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.908.1 Detailed Description

Structure contain the parameter of automatic tracking configuration for the NMEA COM port.

#### Parameters

<i>pbEnabled</i>	<ul style="list-style-type: none"><li>• Automatic XTRA download status<ul style="list-style-type: none"><li>– 0 - Disabled</li><li>– 1 - Enabled</li></ul></li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>
------------------	--

<i>pInterval</i>	<ul style="list-style-type: none"> <li>Interval (hours) between XTRA downloads</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.908.2 Field Documentation

8.908.2.1 `swi_uint256_t unpack_pds_GetXTRAAutomaticDownload_t::ParamPresenceMask`

8.908.2.2 `uint32_t* unpack_pds_GetXTRAAutomaticDownload_t::pbEnabled`

8.908.2.3 `uint16_t* unpack_pds_GetXTRAAutomaticDownload_t::pInterval`

## 8.909 unpack\_pds\_GetXTRANetwork\_t Struct Reference

### Data Fields

- `uint32_t * pPreference`
- `swi_uint256_t ParamPresenceMask`

### 8.909.1 Detailed Description

Structure contain the parameter for XTRA WWAN network preference

#### Parameters

<i>pPreference</i>	<ul style="list-style-type: none"> <li>XTRA WWAN network preference <ul style="list-style-type: none"> <li>0x00 - None (any available network)</li> <li>0x01 - Home-only, only when on home systems</li> <li>0x02 - Roam-only, only when on non-home systems</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.909.2 Field Documentation

8.909.2.1 `swi_uint256_t unpack_pds_GetXTRANetwork_t::ParamPresenceMask`

8.909.2.2 `uint32_t* unpack_pds_GetXTRANetwork_t::pPreference`

## 8.910 unpack\_pds\_GetXTRAVality\_t Struct Reference

### Data Fields

- `uint16_t * pGPSWeek`

- uint16\_t \* [pGPSWeekOffset](#)
- uint16\_t \* [pDuration](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.910.1 Detailed Description

Structure contain the parameter for XTRA database validity period

#### Parameters

<i>pGPSWeek</i>	<ul style="list-style-type: none"> <li>• Starting GPS week of validity period</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pGPSWeek-Offset</i>	<ul style="list-style-type: none"> <li>• Starting GPS week offset (minutes) of validity period</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pDuration</i>	<ul style="list-style-type: none"> <li>• Length of validity period (hours)</li> <li>• NULL pointer - Invalid data.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.910.2 Field Documentation

8.910.2.1 [swi\\_uint256\\_t](#) [unpack\\_pds\\_GetXTRAVValidity\\_t::ParamPresenceMask](#)

8.910.2.2 [uint16\\_t\\*](#) [unpack\\_pds\\_GetXTRAVValidity\\_t::pDuration](#)

8.910.2.3 [uint16\\_t\\*](#) [unpack\\_pds\\_GetXTRAVValidity\\_t::pGPSWeek](#)

8.910.2.4 [uint16\\_t\\*](#) [unpack\\_pds\\_GetXTRAVValidity\\_t::pGPSWeekOffset](#)

## 8.911 unpack\_pds\_SetEventReport\_Ind\_t Struct Reference

### Data Fields

- uint8\_t [has\\_PositionDataNMEA](#)
- char [PositionDataNMEA](#) [200]
- uint8\_t [has\\_dLongitude](#)
- double [dLongitude](#)
- uint8\_t [has\\_dLatitude](#)
- double [dLatitude](#)
- uint8\_t [has\\_SessionStatus](#)
- uint8\_t [SessionStatus](#)
- uint8\_t [has\\_posSrc](#)
- uint32\_t [posSrc](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.911.1 Detailed Description

Structure contain the parameter for the Set event report Indication. Please check has\_<Param\_Name> field for presence of optional parameters

#### Parameters

<i>PositionDataNMEA</i>	<ul style="list-style-type: none"> <li>• Position Data NMEA.</li> <li>• String containing the position data in NMEA sentence format (maximum 200 bytes).</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>dLongitude</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 the Greenwich Meridian.</li> <li>• Units - Decimal degrees</li> <li>• Range - -180 to +180</li> <li>• Note - Value is in double float format</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>dLatitude</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</li> <li>• Note - Value is in double float format</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>session_status</i>	<ul style="list-style-type: none"> <li>• Position Session Status</li> <li>• Status of current session.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Success</li> <li>– 0x01 - In progress</li> <li>– 0x02 - General failure</li> <li>– 0x03 - Timeout</li> <li>– 0x04 - User ended the session</li> <li>– 0x05 - Bad parameter</li> <li>– 0x06 - Phone is offline</li> <li>– 0x07 - Engine is locked</li> <li>– 0x08 - E911 session in progress</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> <li>• Note - This TLV is always sent while a fix is being generated or if it halts due to an error. The Parsed Position Data TLV (0x13) is included if the session status is Success (0x00) or In Progress (0x01); otherwise this TLV is passed alone.</li> </ul>

<i>posSrc</i>	<ul style="list-style-type: none"> <li>• Position Source</li> <li>• This TLV is always sent with the Parsed Position Data TLV (0x13) when the latitude/longitude is marked as valid. Source of the position (bitmask).</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00000001 - GPS</li> <li>– 0x00000002 - Cell ID</li> <li>– 0x00000004 - GLONASS</li> <li>– 0x00000008 - Network</li> <li>– 0x00000010 - External position injection</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.911.2 Field Documentation

- 8.911.2.1 double unpack\_pds\_SetEventReport\_Ind\_t::dLatitude
- 8.911.2.2 double unpack\_pds\_SetEventReport\_Ind\_t::dLongitude
- 8.911.2.3 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::has\_dLatitude
- 8.911.2.4 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::has\_dLongitude
- 8.911.2.5 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::has\_PositionDataNMEA
- 8.911.2.6 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::has\_posSrc
- 8.911.2.7 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::has\_SessionStatus
- 8.911.2.8 swi\_uint256\_t unpack\_pds\_SetEventReport\_Ind\_t::ParamPresenceMask
- 8.911.2.9 char unpack\_pds\_SetEventReport\_Ind\_t::PositionDataNMEA[200]
- 8.911.2.10 uint32\_t unpack\_pds\_SetEventReport\_Ind\_t::posSrc
- 8.911.2.11 uint8\_t unpack\_pds\_SetEventReport\_Ind\_t::SessionStatus

## 8.912 unpack\_pds\_SetPdsState\_Ind\_t Struct Reference

### Data Fields

- uint16\_t [TlvPresent](#)
- uint32\_t [EnabledStatus](#)
- uint32\_t [TrackingStatus](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.912.1 Detailed Description

Structure contain the parameter for the Set PDS State Indication.

## Parameters

<i>EnabledStatus</i>	<ul style="list-style-type: none"> <li>• GPS service state.</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>TrackingStatus</i>	<ul style="list-style-type: none"> <li>• 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.912.2 Field Documentation

8.912.2.1 uint32\_t unpack\_pds\_SetPdsState\_Ind\_t::EnabledStatus

8.912.2.2 swi\_uint256\_t unpack\_pds\_SetPdsState\_Ind\_t::ParamPresenceMask

8.912.2.3 uint16\_t unpack\_pds\_SetPdsState\_Ind\_t::TlvPresent

8.912.2.4 uint32\_t unpack\_pds\_SetPdsState\_Ind\_t::TrackingStatus

## 8.913 unpack\_pds\_SLQSGetAGPSConfig\_t Struct Reference

## Data Fields

- uint32\_t \* [pServerAddress](#)
- uint32\_t \* [pServerPort](#)
- uint8\_t \* [pServerURL](#)
- uint8\_t \* [pServerURLLength](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.913.1 Detailed Description

Structure contain the parameter for PDS AGPS (MS-based) configuration unpack.

## Parameters

<i>pServerAddress</i>	<ul style="list-style-type: none"> <li>• IPv4 address of AGPS server. "0" if not set</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pServerPort</i>	<ul style="list-style-type: none"> <li>• Port number of AGPS server. "0" if not set</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>



<i>pServerURL</i>	<ul style="list-style-type: none"> <li>• URL of the AGPS server. "0" if not set</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul>
<i>pServerURL- Length</i>	<ul style="list-style-type: none"> <li>• URL length of AGPS server. "0" if not set</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.913.2 Field Documentation

8.913.2.1 `swi_uint256_t unpack_pds_SLQSGetAGPSConfig_t::ParamPresenceMask`

8.913.2.2 `uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerAddress`

8.913.2.3 `uint32_t* unpack_pds_SLQSGetAGPSConfig_t::pServerPort`

8.913.2.4 `uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURL`

8.913.2.5 `uint8_t* unpack_pds_SLQSGetAGPSConfig_t::pServerURLLength`

## 8.914 unpack\_pds\_SLQSGetGPSSStateInfo\_t Struct Reference

### Data Fields

- `uint8_t EngineState`
- `uint32_t ValidMask`
- `uint64_t Latitude`
- `uint64_t Longitude`
- `uint32_t HorizontalUncertainty`
- `uint32_t Altitude`
- `uint32_t VerticalUncertainty`
- `uint32_t TimeStmp_tow_ms`
- `uint16_t TimeStmp_gps_week`
- `uint32_t Time_uncert_ms`
- `uint8_t lono_valid`
- `uint32_t gps_ephemeris_sv_msk`
- `uint32_t gps_almanac_sv_msk`
- `uint32_t gps_health_sv_msk`
- `uint32_t gps_visible_sv_msk`
- `uint32_t glo_ephemeris_sv_msk`
- `uint32_t glo_almanac_sv_msk`
- `uint32_t glo_health_sv_msk`
- `uint32_t glo_visible_sv_msk`
- `uint32_t sbas_ephemeris_sv_msk`
- `uint32_t sbas_almanac_sv_msk`
- `uint32_t sbas_health_sv_msk`
- `uint32_t sbas_visible_sv_msk`
- `uint16_t xtra_start_gps_week`

- [uint16\\_t xtra\\_start\\_gps\\_minutes](#)
- [uint16\\_t xtra\\_valid\\_duration\\_hours](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.914.1 Detailed Description

Structure contain the parameter for 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>TimeStmp_tow_ - ms</i>	<ul style="list-style-type: none"> <li>Time stamp in GPS time of week( in milliseconds)</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>TimeStmp_gps_ - week</i>	<ul style="list-style-type: none"> <li>GPS week number</li> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Iono_valid</i>	<ul style="list-style-type: none"> <li>Iono validity.</li> <li>Values: <ul style="list-style-type: none"> <li>0 - Invalid</li> <li>1 - Valid</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>gps_ephemeris_ - sv_msk</i>	<ul style="list-style-type: none"> <li>GPS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>gps_almanac_ - sv_msk</i>	<ul style="list-style-type: none"> <li>GPS SV mask for almanac; if the bit is set, almanac for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>

<i>gps_health_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS SV mask for health; if the bit is set, health for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>gps_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>GPS SV mask for visible Svs; if the bit is set, the SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>glo_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>glo_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS SV mask for almanac; if the bit is set, almanac for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>glo_health_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS SV mask for health; if the bit is set, health for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>glo_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>GLONASS SV mask for visible SVs; if the bit is set, the SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>sbas_ephemeris_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS SV mask for ephemeris; if the bit is set, ephemeris for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>sbas_almanac_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS SV mask for almanac; if the bit is set, almanac for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>sbas_health_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS SV mask for health; if the bit is set, health for that SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>sbas_visible_sv_msk</i>	<ul style="list-style-type: none"> <li>SBAS SV mask for visible SVs; if the bit is set, the SV is available.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
---------------------------	--

## 8.914.2 Field Documentation

- 8.914.2.1 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::Altitude
- 8.914.2.2 uint8\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::EngineState
- 8.914.2.3 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::glo\_almanac\_sv\_msk
- 8.914.2.4 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::glo\_ephemeris\_sv\_msk
- 8.914.2.5 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::glo\_health\_sv\_msk
- 8.914.2.6 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::glo\_visible\_sv\_msk
- 8.914.2.7 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::gps\_almanac\_sv\_msk
- 8.914.2.8 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::gps\_ephemeris\_sv\_msk
- 8.914.2.9 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::gps\_health\_sv\_msk
- 8.914.2.10 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::gps\_visible\_sv\_msk
- 8.914.2.11 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::HorizontalUncertainty
- 8.914.2.12 uint8\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::lono\_valid
- 8.914.2.13 uint64\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::Latitude
- 8.914.2.14 uint64\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::Longitude
- 8.914.2.15 swi\_uint256\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::ParamPresenceMask
- 8.914.2.16 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::sbas\_almanac\_sv\_msk
- 8.914.2.17 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::sbas\_ephemeris\_sv\_msk
- 8.914.2.18 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::sbas\_health\_sv\_msk
- 8.914.2.19 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::sbas\_visible\_sv\_msk
- 8.914.2.20 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::Time\_uncert\_ms
- 8.914.2.21 uint16\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::TimeStmp\_gps\_week
- 8.914.2.22 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::TimeStmp\_tow\_ms
- 8.914.2.23 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::ValidMask
- 8.914.2.24 uint32\_t unpack\_pds\_SLQSGetGPSSStateInfo\_t::VerticalUncertainty

8.914.2.25 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_start_gps_minutes`

8.914.2.26 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_start_gps_week`

8.914.2.27 `uint16_t unpack_pds_SLQSGetGPSStateInfo_t::extra_valid_duration_hours`

## 8.915 `unpack_qmi_t` Struct Reference

### Data Fields

- enum [msgtype](#) `type`
- `uint16_t` [msgid](#)
- `uint16_t` [xid](#)

### 8.915.1 Detailed Description

qmi response context

#### Parameters

<code>out</code>	<i>type</i>	message type
<code>out</code>	<i>msgid</i>	message id
<code>out</code>	<i>xid</i>	transaction id

### 8.915.2 Field Documentation

8.915.2.1 `uint16_t unpack_qmi_t::msgid`

8.915.2.2 `enum msgtype unpack_qmi_t::type`

8.915.2.3 `uint16_t unpack_qmi_t::xid`

## 8.916 `unpack_qos_BindDataPort_t` Struct Reference

### Data Fields

- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.916.1 Detailed Description

This structure contains unpack Qos BindDataPort.

#### Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.916.2 Field Documentation

8.916.2.1 `swi_uint256_t unpack_qos_BindDataPort_t::ParamPresenceMask`

## 8.917 unpack\_qos\_dataRate\_t Struct Reference

### Data Fields

- uint32\_t [dataRateMax](#)
- uint32\_t [guaranteedRate](#)

### 8.917.1 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)

### 8.917.2 Field Documentation

8.917.2.1 uint32\_t unpack\_qos\_dataRate\_t::dataRateMax

8.917.2.2 uint32\_t unpack\_qos\_dataRate\_t::guaranteedRate

## 8.918 unpack\_qos\_IPv4Addr\_t Struct Reference

### Data Fields

- uint32\_t [addr](#)
- uint32\_t [subnetMask](#)

### 8.918.1 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>

### 8.918.2 Field Documentation

8.918.2.1 uint32\_t unpack\_qos\_IPv4Addr\_t::addr

8.918.2.2 uint32\_t unpack\_qos\_IPv4Addr\_t::subnetMask

## 8.919 unpack\_qos\_IPv6Addr\_t Struct Reference

### Data Fields

- uint8\_t [addr](#) [16]

- uint8\_t [prefixLen](#)

### 8.919.1 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.

### 8.919.2 Field Documentation

8.919.2.1 uint8\_t unpack\_qos\_IPv6Addr\_t::addr[16]

8.919.2.2 uint8\_t unpack\_qos\_IPv6Addr\_t::prefixLen

## 8.920 unpack\_qos\_IPv6TrafCls\_t Struct Reference

### Data Fields

- uint8\_t [val](#)
- uint8\_t [mask](#)

### 8.920.1 Detailed Description

This structure contains the IPv6 filter traffic class

#### Parameters

<i>val</i>	The traffic class value
<i>mask</i>	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 & IPv6_filter_traffic_class_mask) Example: <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>

### 8.920.2 Field Documentation

8.920.2.1 uint8\_t unpack\_qos\_IPv6TrafCls\_t::mask

8.920.2.2 uint8\_t unpack\_qos\_IPv6TrafCls\_t::val

## 8.921 unpack\_qos\_pktErrRate\_t Struct Reference

### Data Fields

- uint16\_t [multiplier](#)



- uint16\_t [exponent](#)

### 8.921.1 Detailed Description

This structure contains the IP flow packet error rate

#### Parameters

<i>multiplier</i>	Factor m in calculating packet error rate: $E = m \cdot 10^{(-p)}$
<i>exponent</i>	Factor p in calculating packet error rate (see above)

### 8.921.2 Field Documentation

8.921.2.1 uint16\_t unpack\_qos\_pktErrRate\_t::exponent

8.921.2.2 uint16\_t unpack\_qos\_pktErrRate\_t::multiplier

## 8.922 unpack\_qos\_Port\_t Struct Reference

### Data Fields

- uint16\_t [port](#)
- uint16\_t [range](#)

### 8.922.1 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

### 8.922.2 Field Documentation

8.922.2.1 uint16\_t unpack\_qos\_Port\_t::port

8.922.2.2 uint16\_t unpack\_qos\_Port\_t::range

## 8.923 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](#)

### 8.923.1 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="#">LITEQMI_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="#">LITEQMI_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>

### 8.923.2 Field Documentation

8.923.2.1 `uint8_t unpack_qos_QosFlowInfo_t::BearerID`

8.923.2.2 `uint8_t unpack_qos_QosFlowInfo_t::is_RxQFlowGranted_Available`

8.923.2.3 `uint8_t unpack_qos_QosFlowInfo_t::is_TxQFlowGranted_Available`

- 8.923.2.4 `uint8_t unpack_qos_QosFlowInfo_t::NumRxFilters`
- 8.923.2.5 `uint8_t unpack_qos_QosFlowInfo_t::NumTxFilters`
- 8.923.2.6 `unpack_qos_QosFlowInfoState_t unpack_qos_QosFlowInfo_t::QFlowState`
- 8.923.2.7 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::RxQFilter[25]`
- 8.923.2.8 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::RxQFlowGranted`
- 8.923.2.9 `unpack_qos_swiQosFilter_t unpack_qos_QosFlowInfo_t::TxQFilter[25]`
- 8.923.2.10 `unpack_qos_swiQosFlow_t unpack_qos_QosFlowInfo_t::TxQFlowGranted`

## 8.924 unpack\_qos\_QosFlowInfoState\_t Struct Reference

### Data Fields

- `uint32_t id`
- `uint8_t isNewFlow`
- `uint8_t state`

### 8.924.1 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>

### 8.924.2 Field Documentation

- 8.924.2.1 `uint32_t unpack_qos_QosFlowInfoState_t::id`
- 8.924.2.2 `uint8_t unpack_qos_QosFlowInfoState_t::isNewFlow`
- 8.924.2.3 `uint8_t unpack_qos_QosFlowInfoState_t::state`

## 8.925 unpack\_qos\_SLQSQosGetNetworkStatus\_t Struct Reference

## Data Fields

- uint8\_t [NWQoSStatus](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.925.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.925.2 Field Documentation

8.925.2.1 uint8\_t [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t::NWQoSStatus](#)

8.925.2.2 swi\_uint256\_t [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t::ParamPresenceMask](#)

## 8.926 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](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.926.1 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> <li>• Bit to check in ParamPresenceMask - 3</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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.926.2 Field Documentation

8.926.2.1 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl`

8.926.2.2 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl_ext`

8.926.2.3 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_dl_ext2`

8.926.2.4 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul`

8.926.2.5 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul_ext`

8.926.2.6 `uint8_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ambr_ul_ext2`

8.926.2.7 `uint32_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::apnId`

8.926.2.8 `swi_uint256_t unpack_qos_SLQSQoSwiReadApnExtraParams_t::ParamPresenceMask`

## 8.927 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]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.927.1 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> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_tx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets sent</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_tx_pkt_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) packets dropped</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_rx_pkt</i>	<ul style="list-style-type: none"> <li>• sum of all packets received</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_tx_bytes</i>	<ul style="list-style-type: none"> <li>• sum of all bytes sent</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_tx_bytes_drp</i>	<ul style="list-style-type: none"> <li>• sum of all(TX) bytes dropped</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>total_rx_bytes</i>	<ul style="list-style-type: none"> <li>• number of received bytes for the QoS flow ID</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>numQosFlow</i>	<ul style="list-style-type: none"> <li>• pointer to number of QoS flow Stat</li> <li>• Bit to check in ParamPresenceMask - <b>4</b></li> </ul>

<i>qosFlow[LITEQMI_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="#">LITEQMI_MAX_QOS_FLOW_PER_APN_STATS</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 4</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.927.2 Field Documentation

8.927.2.1 `uint32_t unpack_qos_SLQSQosSviReadDataStats_t::apnId`

8.927.2.2 `uint32_t unpack_qos_SLQSQosSviReadDataStats_t::numQosFlow`

8.927.2.3 `swi_uint256_t unpack_qos_SLQSQosSviReadDataStats_t::ParamPresenceMask`

8.927.2.4 `unpack_QosFlowStat_t unpack_qos_SLQSQosSviReadDataStats_t::qosFlow[10]`

8.927.2.5 `uint64_t unpack_qos_SLQSQosSviReadDataStats_t::total_rx_bytes`

8.927.2.6 `uint32_t unpack_qos_SLQSQosSviReadDataStats_t::total_rx_pkt`

8.927.2.7 `uint64_t unpack_qos_SLQSQosSviReadDataStats_t::total_tx_bytes`

8.927.2.8 `uint64_t unpack_qos_SLQSQosSviReadDataStats_t::total_tx_bytes_drp`

8.927.2.9 `uint32_t unpack_qos_SLQSQosSviReadDataStats_t::total_tx_pkt`

8.927.2.10 `uint32_t unpack_qos_SLQSQosSviReadDataStats_t::total_tx_pkt_drp`

## 8.928 unpack\_qos\_SLQSSetQosEventCallback\_ind\_t Struct Reference

### Data Fields

- `uint8_t NumFlows`
- `unpack_qos_QosFlowInfo_t QosFlowInfo [8]`
- `swi_uint256_t ParamPresenceMask`

### 8.928.1 Detailed Description

Structure with QoS event details

#### Parameters

<i>NumFlows</i>	<ul style="list-style-type: none"> <li>• Number of QoS flows available</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
-----------------	---

<i>QosFlowInfo</i>	<ul style="list-style-type: none"> <li>• The Qos flow details, please check <a href="#">unpack_qos_QosFlowInfo_t</a> for more information</li> <li>• See <a href="#">LITEQMI_MAX_QOS_FLOWS</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.928.2 Field Documentation

8.928.2.1 `uint8_t unpack_qos_SLQSSetQosEventCallback_ind_t::NumFlows`

8.928.2.2 `swi_uint256_t unpack_qos_SLQSSetQosEventCallback_ind_t::ParamPresenceMask`

8.928.2.3 `unpack_qos_QosFlowInfo_t unpack_qos_SLQSSetQosEventCallback_ind_t::QosFlowInfo[8]`

## 8.929 unpack\_qos\_SLQSSetQosEventCallback\_t Struct Reference

### Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.929.1 Detailed Description

This structure contains unpack QOS SLQSSetQosEventCallback. param ParamPresenceMask

- bitmask representation to indicate valid parameters.

### 8.929.2 Field Documentation

8.929.2.1 `swi_uint256_t unpack_qos_SLQSSetQosEventCallback_t::ParamPresenceMask`

## 8.930 unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t Struct Reference

### Data Fields

- `uint8_t status`
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.930.1 Detailed Description

Structure with network's QoS status

#### Parameters

<i>status</i>	<p>Network QoS support status</p> <ul style="list-style-type: none"> <li>• 0x00 - Current network does not support QoS</li> <li>• 0x01 - Current network supports QoS</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
---------------	--



<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## Note

- Technology Supported: CDMA

## 8.930.2 Field Documentation

8.930.2.1 `swi_uint256_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::ParamPresenceMask`

8.930.2.2 `uint8_t unpack_qos_SLQSSetQosNWStatusCallback_ind_t::status`

## 8.931 unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t Struct Reference

## Data Fields

- `uint16_t event`
- `swi_uint256_t ParamPresenceMask`

## 8.931.1 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> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.931.2 Field Documentation

8.931.2.1 `uint16_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::event`

8.931.2.2 `swi_uint256_t unpack_qos_SLQSSetQosPriEventCallback_ind_t::ParamPresenceMask`

## 8.932 unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t Struct Reference

## Data Fields

- `uint32_t id`
- `uint8_t status`
- `uint8_t event`
- `uint8_t reason`
- `swi_uint256_t ParamPresenceMask`

### 8.932.1 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.932.2 Field Documentation

8.932.2.1 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::event`

8.932.2.2 `uint32_t unpack_qos_SLQSSetQosStatusCallback_ind_t::id`

8.932.2.3 `swi_uint256_t unpack_qos_SLQSSetQosStatusCallback_ind_t::ParamPresenceMask`

8.932.2.4 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::reason`

8.932.2.5 `uint8_t unpack_qos_SLQSSetQosStatusCallback_ind_t::status`

## 8.933 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_TranDstPort_Available`
- `unpack_qos_Port_t TranDstPort`

### 8.933.1 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 soruce 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 soruce 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 protocol filter 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>

### 8.933.2 Field Documentation

8.933.2.1 `uint32_t unpack_qos_swiQosFilter_t::EspSpi`

8.933.2.2 `uint16_t unpack_qos_swiQosFilter_t::Id`

8.933.2.3 `uint8_t unpack_qos_swiQosFilter_t::index`

8.933.2.4 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4DstAddr`

8.933.2.5 `unpack_qos_IPv4Addr_t unpack_qos_swiQosFilter_t::IPv4SrcAddr`

8.933.2.6 `unpack_qos_Tos_t unpack_qos_swiQosFilter_t::IPv4Tos`

8.933.2.7 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6DstAddr`

8.933.2.8 `uint32_t unpack_qos_swiQosFilter_t::IPv6Label`

8.933.2.9 `unpack_qos_IPv6Addr_t unpack_qos_swiQosFilter_t::IPv6SrcAddr`

8.933.2.10 `unpack_qos_IPv6TrafCls_t unpack_qos_swiQosFilter_t::IPv6TrafCls`

8.933.2.11 `uint8_t unpack_qos_swiQosFilter_t::is_EspSpi_Available`

8.933.2.12 `uint8_t unpack_qos_swiQosFilter_t::is_Id_Available`

8.933.2.13 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4DstAddr_Available`

8.933.2.14 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4SrcAddr_Available`

8.933.2.15 `uint8_t unpack_qos_swiQosFilter_t::is_IPv4Tos_Available`

8.933.2.16 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6DstAddr_Available`

8.933.2.17 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6Label_Available`

8.933.2.18 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6SrcAddr_Available`

8.933.2.19 `uint8_t unpack_qos_swiQosFilter_t::is_IPv6TrafCls_Available`

8.933.2.20 `uint8_t unpack_qos_swiQosFilter_t::is_NxtHdrProto_Available`

8.933.2.21 `uint8_t unpack_qos_swiQosFilter_t::is_Precedence_Available`

8.933.2.22 `uint8_t unpack_qos_swiQosFilter_t::is_TCPDstPort_Available`

8.933.2.23 `uint8_t unpack_qos_swiQosFilter_t::is_TCPSrcPort_Available`

8.933.2.24 `uint8_t unpack_qos_swiQosFilter_t::is_TranDstPort_Available`

8.933.2.25 `uint8_t unpack_qos_swiQosFilter_t::is_TranSrcPort_Available`

8.933.2.26 `uint8_t unpack_qos_swiQosFilter_t::is_UDPDstPort_Available`

8.933.2.27 `uint8_t unpack_qos_swiQosFilter_t::is_UDPSrcPort_Available`

8.933.2.28 `uint8_t unpack_qos_swiQosFilter_t::NxtHdrProto`

8.933.2.29 `uint16_t unpack_qos_swiQosFilter_t::Precedence`

8.933.2.30 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPDstPort`

8.933.2.31 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TCPSrcPort`

8.933.2.32 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranDstPort`

8.933.2.33 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::TranSrcPort`

8.933.2.34 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPDstPort`

8.933.2.35 `unpack_qos_Port_t unpack_qos_swiQosFilter_t::UDPSrcPort`

8.933.2.36 `uint8_t unpack_qos_swiQosFilter_t::version`

## 8.934 `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](#)

### 8.934.1 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>

## 8.934.2 Field Documentation

8.934.2.1 `unpack_qos_dataRate_t` `unpack_qos_swiQosFlow_t::DataRate`

8.934.2.2 `uint8_t` `unpack_qos_swiQosFlow_t::index`

8.934.2.3 `uint8_t` `unpack_qos_swiQosFlow_t::is_DataRate_Available`

8.934.2.4 `uint8_t` `unpack_qos_swiQosFlow_t::is_Jitter_Available`

8.934.2.5 `uint8_t` `unpack_qos_swiQosFlow_t::is_Latency_Available`

8.934.2.6 `uint8_t` `unpack_qos_swiQosFlow_t::is_LteQci_Available`

8.934.2.7 `uint8_t` `unpack_qos_swiQosFlow_t::is_MaxAllowedPktSz_Available`

8.934.2.8 `uint8_t` `unpack_qos_swiQosFlow_t::is_MinPolicedPktSz_Available`

8.934.2.9 `uint8_t` `unpack_qos_swiQosFlow_t::is_PktErrRate_Available`

8.934.2.10 `uint8_t` `unpack_qos_swiQosFlow_t::is_ProfileId3GPP2_Available`

8.934.2.11 `uint8_t` `unpack_qos_swiQosFlow_t::is-TokenBucket_Available`

8.934.2.12 `uint8_t` `unpack_qos_swiQosFlow_t::is_TrafficClass_Available`

8.934.2.13 `uint8_t` `unpack_qos_swiQosFlow_t::is_val_3GPP2Pri_Available`

8.934.2.14 `uint8_t` `unpack_qos_swiQosFlow_t::is_val_3GPPImCn_Available`

- 8.934.2.15 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPResResidualBER_Available`
- 8.934.2.16 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPSigInd_Available`
- 8.934.2.17 `uint8_t unpack_qos_swiQosFlow_t::is_val_3GPPTraHdlPri_Available`
- 8.934.2.18 `uint32_t unpack_qos_swiQosFlow_t::Jitter`
- 8.934.2.19 `uint32_t unpack_qos_swiQosFlow_t::Latency`
- 8.934.2.20 `uint8_t unpack_qos_swiQosFlow_t::LteQci`
- 8.934.2.21 `uint32_t unpack_qos_swiQosFlow_t::MaxAllowedPktSz`
- 8.934.2.22 `uint32_t unpack_qos_swiQosFlow_t::MinPolicedPktSz`
- 8.934.2.23 `unpack_qos_pktErrRate_t unpack_qos_swiQosFlow_t::PktErrRate`
- 8.934.2.24 `uint16_t unpack_qos_swiQosFlow_t::ProfileId3GPP2`
- 8.934.2.25 `unpack_qos_tokenBucket_t unpack_qos_swiQosFlow_t::TokenBucket`
- 8.934.2.26 `uint8_t unpack_qos_swiQosFlow_t::TrafficClass`
- 8.934.2.27 `uint8_t unpack_qos_swiQosFlow_t::val_3GPP2Pri`
- 8.934.2.28 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPIImCn`
- 8.934.2.29 `uint16_t unpack_qos_swiQosFlow_t::val_3GPPResResidualBER`
- 8.934.2.30 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPSigInd`
- 8.934.2.31 `uint8_t unpack_qos_swiQosFlow_t::val_3GPPTraHdlPri`

## 8.935 `unpack_qos_tokenBucket_t` Struct Reference

### Data Fields

- `uint32_t` [peakRate](#)
- `uint32_t` [tokenRate](#)
- `uint32_t` [bucketSz](#)

### 8.935.1 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

### 8.935.2 Field Documentation

8.935.2.1 uint32\_t unpack\_qos\_tokenBucket\_t::bucketSz

8.935.2.2 uint32\_t unpack\_qos\_tokenBucket\_t::peakRate

8.935.2.3 uint32\_t unpack\_qos\_tokenBucket\_t::tokenRate

## 8.936 unpack\_qos\_Tos\_t Struct Reference

### Data Fields

- uint8\_t [val](#)
- uint8\_t [mask](#)

### 8.936.1 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>

### 8.936.2 Field Documentation

8.936.2.1 uint8\_t unpack\_qos\_Tos\_t::mask

8.936.2.2 uint8\_t unpack\_qos\_Tos\_t::val

## 8.937 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](#)

### 8.937.1 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>

## 8.937.2 Field Documentation

8.937.2.1 `uint32_t unpack_QosFlowStat_t::bearerId`8.937.2.2 `uint64_t unpack_QosFlowStat_t::tx_bytes`8.937.2.3 `uint64_t unpack_QosFlowStat_t::tx_bytes_drp`8.937.2.4 `uint32_t unpack_QosFlowStat_t::tx_pkt`8.937.2.5 `uint32_t unpack_QosFlowStat_t::tx_pkt_drp`8.938 `unpack_result_t` Struct Reference

## Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.938.1 Detailed Description

unpack result struct

## Parameters

<code>out</code>	<i>Tlvresult</i>	
<code>out</code>	<i>ParamPresenceMask</i>	Parameter presence bit mask

## 8.938.2 Field Documentation

8.938.2.1 `swi_uint256_t unpack_result_t::ParamPresenceMask`8.938.2.2 `uint16_t unpack_result_t::Tlvresult`

## 8.939 unpack\_rms\_GetSMSWake\_t Struct Reference

### Data Fields

- uint32\_t [enabled](#)
- uint32\_t [wake\\_mask](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.939.1 Detailed Description

Unpack get SMS wake parameters

#### Parameters

<i>enabled</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>wake_mask</i>	<ul style="list-style-type: none"> <li>• SMS wake mask to search for incoming messages (only relevant when enabled)</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.939.2 Field Documentation

8.939.2.1 uint32\_t unpack\_rms\_GetSMSWake\_t::enabled

8.939.2.2 swi\_uint256\_t unpack\_rms\_GetSMSWake\_t::ParamPresenceMask

8.939.2.3 uint32\_t unpack\_rms\_GetSMSWake\_t::wake\_mask

## 8.940 unpack\_rms\_SetSMSWake\_t Struct Reference

### Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.940.1 Detailed Description

This structure contains unpack Set SMS Wake.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
---------------------------	--

## 8.940.2 Field Documentation

8.940.2.1 `swi_uint256_t unpack_rms_SetSMSWake_t::ParamPresenceMask`

## 8.941 `unpack_RMTransferStatistics_ind_t` Struct Reference

### Data Fields

- [wds\\_DataULongTlv TxOkConutTlv](#)
- [wds\\_DataULongTlv RxOkConutTlv](#)
- [wds\\_DataULongLongTlv TxOkByteCountTlv](#)
- [wds\\_DataULongLongTlv RxOkByteCountTlv](#)
- [wds\\_DataULongTlv TxDropConutTlv](#)
- [wds\\_DataULongTlv RxDropConutTlv](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.941.1 Detailed Description

WDS packet 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>RxOkConutTlv</i>	<ul style="list-style-type: none"> <li>• Rx Ok Packet Tlv Value.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>TxOkByteCount-Tlv</i>	<ul style="list-style-type: none"> <li>• Tx Ok Byte Count Packet Tlv Value.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>RxOkByteCount-Tlv</i>	<ul style="list-style-type: none"> <li>• Rx Ok Byte Count Packet Tlv Value.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>TxDropConutTlv</i>	<ul style="list-style-type: none"> <li>• Tx Drop Count Packet Tlv Value.</li> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>RxDropConutTlv</i>	<ul style="list-style-type: none"> <li>• Rx Drop Count Packet Tlv Value.</li> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>

### 8.941.2 Field Documentation

8.941.2.1 `swi_uint256_t unpack_RMTransferStatistics_ind_t::ParamPresenceMask`

8.941.2.2 `wds_DataULongTlv unpack_RMTransferStatistics_ind_t::RxDropConutTlv`

8.941.2.3 wds\_DataULongLongTlv unpack\_RMTransferStatistics\_ind\_t::RxOkByteCountTlv

8.941.2.4 wds\_DataULongTlv unpack\_RMTransferStatistics\_ind\_t::RxOkConutTlv

8.941.2.5 wds\_DataULongTlv unpack\_RMTransferStatistics\_ind\_t::TxDropConutTlv

8.941.2.6 wds\_DataULongLongTlv unpack\_RMTransferStatistics\_ind\_t::TxOkByteCountTlv

8.941.2.7 wds\_DataULongTlv unpack\_RMTransferStatistics\_ind\_t::TxOkConutTlv

## 8.942 unpack\_sar\_SLQSGetRfSarState\_t Struct Reference

### Data Fields

- [uint32\\_t](#) \* [pSarRFState](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.942.1 Detailed Description

This structure contains unpack get RF SAR state parameter.

#### 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> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.942.2 Field Documentation

8.942.2.1 [swi\\_uint256\\_t](#) [unpack\\_sar\\_SLQSGetRfSarState\\_t::ParamPresenceMask](#)

8.942.2.2 [uint32\\_t](#)\* [unpack\\_sar\\_SLQSGetRfSarState\\_t::pSarRFState](#)

## 8.943 unpack\_sms\_GetSMSCAddress\_t Struct Reference

### Data Fields

- [uint8\\_t](#) [addressSize](#)
- [uint8\\_t](#) \* [pSMSCAddress](#)

- [uint8\\_t typeSize](#)
- [uint8\\_t \\* pSMSCType](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.943.1 Detailed Description

Structure contains Gets the SMS center address parameters.

#### 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pSMSCAddress</i>	<ul style="list-style-type: none"> <li>• The SMS center address represented as a NULL terminated string.</li> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>pSMSCType</i>	<ul style="list-style-type: none"> <li>• The SMS center address type represented as a NULL terminated string.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.943.2 Field Documentation

8.943.2.1 [uint8\\_t unpack\\_sms\\_GetSMSCAddress\\_t::addressSize](#)

8.943.2.2 [swi\\_uint256\\_t unpack\\_sms\\_GetSMSCAddress\\_t::ParamPresenceMask](#)

8.943.2.3 [uint8\\_t\\* unpack\\_sms\\_GetSMSCAddress\\_t::pSMSCAddress](#)

8.943.2.4 [uint8\\_t\\* unpack\\_sms\\_GetSMSCAddress\\_t::pSMSCType](#)

8.943.2.5 [uint8\\_t unpack\\_sms\\_GetSMSCAddress\\_t::typeSize](#)

## 8.944 unpack\_sms\_SaveSMS\_t Struct Reference

#### Data Fields

- [uint32\\_t \\* pMessageIndex](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)



### 8.944.1 Detailed Description

This structure contains unpack save SMS parameter.

#### Parameters

<i>pMessageIndex</i>	<ul style="list-style-type: none"> <li>The message index assigned by the device</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.944.2 Field Documentation

8.944.2.1 `swi_uint256_t unpack_sms_SaveSMS_t::ParamPresenceMask`

8.944.2.2 `uint32_t* unpack_sms_SaveSMS_t::pMessageIndex`

## 8.945 unpack\_sms\_SendSMS\_t Struct Reference

### Data Fields

- `uint16_t` [messageID](#)
- `uint32_t` [messageFailureCode](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.945.1 Detailed Description

This structure contains unpack send SMS parameters.

#### Parameters

<i>messageID</i>	<ul style="list-style-type: none"> <li>WMS message ID</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>messageFailureCode</i>	<ul style="list-style-type: none"> <li>pointer to message failure code.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.945.2 Field Documentation

8.945.2.1 `uint32_t unpack_sms_SendSMS_t::messageFailureCode`

8.945.2.2 `uint16_t unpack_sms_SendSMS_t::messageID`

8.945.2.3 `swi_uint256_t unpack_sms_SendSMS_t::ParamPresenceMask`

## 8.946 unpack\_sms\_SetNewSMSCallback\_ind\_t Struct Reference

### Data Fields

- [newMTMessageTlv](#) NewMMTlv
- [transferRouteMessageTlv](#) TRMessageTlv
- [messageModeTlv](#) MMTlv
- [sMSEtwsMessageTlv](#) ETWSTlv
- [eTWSPLMNInfoTlv](#) ETWSPLMNTlv
- [sMSCAddressTlv](#) SMSCTlv
- [sMSOnIMSTlv](#) IMSTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.946.1 Detailed Description

This structure contains unpack new SMS callback indication.

#### Parameters

<i>NewMMTlv</i>	<ul style="list-style-type: none"> <li>• MT message</li> <li>• See <a href="#">newMTMessageTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>SMSCTlv</i>	<ul style="list-style-type: none"> <li>• SMSC Address</li> <li>• See <a href="#">sMSCAddressTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></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> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

## 8.946.2 Field Documentation

8.946.2.1 `eTWSPLMNInfoTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSPLMNTlv`

8.946.2.2 `sMSEtwsMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::ETWSTlv`

8.946.2.3 `sMSOnIMSTlv` `unpack_sms_SetNewSMSCallback_ind_t::IMSTlv`

8.946.2.4 `messageModeTlv` `unpack_sms_SetNewSMSCallback_ind_t::MMTlv`

8.946.2.5 `newMTMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::NewMMTlv`

8.946.2.6 `swi_uint256_t` `unpack_sms_SetNewSMSCallback_ind_t::ParamPresenceMask`

8.946.2.7 `sMSCAddressTlv` `unpack_sms_SetNewSMSCallback_ind_t::SMSCtlv`

8.946.2.8 `transferRouteMessageTlv` `unpack_sms_SetNewSMSCallback_ind_t::TRMessageTlv`

## 8.947 unpack\_sms\_SetNewSMSCallback\_t Struct Reference

### Data Fields

- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.947.1 Detailed Description

This structure contains unpack set new SMS callback.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

## 8.947.2 Field Documentation

8.947.2.1 `swi_uint256_t` `unpack_sms_SetNewSMSCallback_t::ParamPresenceMask`

## 8.948 unpack\_sms\_SetSMSCAddress\_t Struct Reference

### Data Fields

- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.948.1 Detailed Description

This structure contains unpack Set SMS Address parameter.

## Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.948.2 Field Documentation

8.948.2.1 `swi_uint256_t` `unpack_sms_SetSMSCAddress_t::ParamPresenceMask`

8.949 `unpack_sms_SLQSDeleteSMS_t` Struct Reference

## Data Fields

- [swi\\_uint256\\_t](#) `ParamPresenceMask`

## 8.949.1 Detailed Description

This structure contains unpack delete SMS parameters.

## Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.949.2 Field Documentation

8.949.2.1 `swi_uint256_t` `unpack_sms_SLQSDeleteSMS_t::ParamPresenceMask`

8.950 `unpack_sms_SLQSGetIndicationRegister_t` Struct Reference

## Data Fields

- [sms\\_getIndicationReg](#) \* `pGetIndicationRegInfo`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

## 8.950.1 Detailed Description

Structure containing Parameters that provides registration state of different WMS indications.

## Parameters

<i>pGetIndicationRegInfo</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <code>getIndicationReg</code> <ul style="list-style-type: none"> <li>See <a href="#">sms_getIndicationReg</a> for more information</li> <li><code>pRegTransLayerInfoEvt</code> <ul style="list-style-type: none"> <li>Bit to check in <code>ParamPresenceMask</code> - <b>16</b></li> </ul> </li> <li><code>pRegTransNWRegInfoEvt</code> <ul style="list-style-type: none"> <li>Bit to check in <code>ParamPresenceMask</code> - <b>17</b></li> </ul> </li> <li><code>pRegCallStatInfoEvt</code> <ul style="list-style-type: none"> <li>Bit to check in <code>ParamPresenceMask</code> - <b>18</b></li> </ul> </li> </ul> </li> </ul>
------------------------------	--

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.950.2 Field Documentation

8.950.2.1 `swi_uint256_t unpack_sms_SLQSGetIndicationRegister_t::ParamPresenceMask`

8.950.2.2 `sms_getIndicationReg* unpack_sms_SLQSGetIndicationRegister_t::pGetIndicationRegInfo`

## 8.951 unpack\_sms\_SLQSGetMessageWaiting\_t Struct Reference

### Data Fields

- [sms\\_getMsgWaitingInfo](#) \* [pGetMsgWaitingInfoResp](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.951.1 Detailed Description

Structure contain Parameter that provide information about the message waiting information.

#### Parameters

<i>pGetMsgWaitingInfoResp</i>	<ul style="list-style-type: none"> <li>Pointer to structure of <a href="#">getMsgWaitingInfoResp</a> <ul style="list-style-type: none"> <li>See <a href="#">sms_getMsgWaitingInfo</a> for more information</li> </ul> </li> <li>Bit to check in <a href="#">ParamPresenceMask</a> - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.951.2 Field Documentation

8.951.2.1 `swi_uint256_t unpack_sms_SLQSGetMessageWaiting_t::ParamPresenceMask`

8.951.2.2 `sms_getMsgWaitingInfo* unpack_sms_SLQSGetMessageWaiting_t::pGetMsgWaitingInfoResp`

## 8.952 unpack\_sms\_SLQSGetSMS\_t Struct Reference

### Data Fields

- `uint32_t` [messageTag](#)
- `uint32_t` [messageFormat](#)
- `uint32_t` [messageSize](#)
- `uint8_t` [message](#) [2048]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.952.1 Detailed Description

This sturcture contains unpack get SMS parameters.

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

- Bit to check in ParamPresenceMask - 1

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

- Bit to check in ParamPresenceMask - 1

## Parameters

<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> <li>• Upon successful output the actual number of bytes written to the message array.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>message</i>	<ul style="list-style-type: none"> <li>• The message contents array</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.952.2 Field Documentation

8.952.2.1 `uint8_t unpack_sms_SLQSGetSMS_t::message[2048]`

8.952.2.2 `uint32_t unpack_sms_SLQSGetSMS_t::messageFormat`

8.952.2.3 `uint32_t unpack_sms_SLQSGetSMS_t::messageSize`

8.952.2.4 `uint32_t unpack_sms_SLQSGetSMS_t::messageTag`

8.952.2.5 `swi_uint256_t unpack_sms_SLQSGetSMS_t::ParamPresenceMask`

8.953 `unpack_sms_SLQSGetSmsBroadcastConfig_t` Struct Reference

## Data Fields

- [sms\\_qaQmi3GPPBroadcastCfgInfo](#) \* [pBroadcastConfig](#)
- [sms\\_qaQmi3GPP2BroadcastCfgInfo](#) \* [pCDMABroadcastConfig](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.953.1 Detailed Description

Structure contain parameters that Provides Information about the SMS BroadcastConfiguration

#### Parameters

<i>pBroadcast-Config</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP Broadcast Information(Optional).</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pCDMA-BroadcastConfig</i>	<ul style="list-style-type: none"> <li>• The data for 3GPP2 Broadcast Information(Optional).</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.953.2 Field Documentation

8.953.2.1 [swi\\_uint256\\_t](#) [unpack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t::ParamPresenceMask](#)

8.953.2.2 [sms\\_qaQmi3GPPBroadcastCfgInfo](#)\* [unpack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t::pBroadcastConfig](#)

8.953.2.3 [sms\\_qaQmi3GPP2BroadcastCfgInfo](#)\* [unpack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t::pCDMABroadcastConfig](#)

## 8.954 unpack\_sms\_SLQSGetSMSList\_t Struct Reference

## Data Fields

- [uint32\\_t](#) [messageListSize](#)
- [qmiSmsMessageList](#) [messageList](#) [255]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.954.1 Detailed Description

This structure contains unpack get SMS list parameters.

#### 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.954.2 Field Documentation

8.954.2.1 `qmiSmsMessageList unpack_sms_SLQSGetSMSList_t::messageList[255]`

8.954.2.2 `uint32_t unpack_sms_SLQSGetSMSList_t::messageListSize`

8.954.2.3 `swi_uint256_t unpack_sms_SLQSGetSMSList_t::ParamPresenceMask`

## 8.955 unpack\_sms\_SLQSGetTransLayerInfo\_t Struct Reference

### Data Fields

- [sms\\_getTransLayerInfo](#) \* [pGetTransLayerInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.955.1 Detailed Description

Structure contain parameters that gives information about the transport layer.

#### Parameters

<i>pGetTransLayerInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of getTransLayerInfo <ul style="list-style-type: none"> <li>– See <a href="#">sms_getTransLayerInfo</a> for more information</li> <li>– <code>pGetTransLayerInfo-&gt;pRegInd</code> <ul style="list-style-type: none"> <li>* Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> <li>– <code>pGetTransLayerInfo-&gt;pTransLayerInfo</code> <ul style="list-style-type: none"> <li>* Bit to check in ParamPresenceMask - <b>17</b></li> </ul> </li> </ul> </li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.955.2 Field Documentation

8.955.2.1 `swi_uint256_t unpack_sms_SLQSGetTransLayerInfo_t::ParamPresenceMask`

8.955.2.2 `sms_getTransLayerInfo* unpack_sms_SLQSGetTransLayerInfo_t::pGetTransLayerInfo`

## 8.956 unpack\_sms\_SLQSGetTransNWRegInfo\_t Struct Reference



## Data Fields

- [sms\\_getTransNWRegInfo](#) \* [pGetTransNWRegInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.956.1 Detailed Description

Structure containing parameters that provides transport layer network registration info.

## Parameters

<i>pGetTransNW-RegInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <code>getTransNWRegInfo</code> <ul style="list-style-type: none"> <li>– See <a href="#">sms_getTransNWRegInfo</a> for more information</li> <li>– Bit to check in <code>ParamPresenceMask</code> - <b>16</b></li> </ul> </li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.956.2 Field Documentation

8.956.2.1 `swi_uint256_t` `unpack_sms_SLQSGetTransNWRegInfo_t::ParamPresenceMask`

8.956.2.2 `sms_getTransNWRegInfo*` `unpack_sms_SLQSGetTransNWRegInfo_t::pGetTransNWRegInfo`

## 8.957 unpack\_sms\_SLQSModifySMSStatus\_t Struct Reference

## Data Fields

- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.957.1 Detailed Description

This structure contains unpack modify SMS status parameters.

## Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
---------------------------	--

## 8.957.2 Field Documentation

8.957.2.1 `swi_uint256_t` `unpack_sms_SLQSModifySMSStatus_t::ParamPresenceMask`

## 8.958 unpack\_sms\_SLQSNWRegInfoCallback\_ind\_t Struct Reference

## Data Fields

- `uint8_t` [NWRegStat](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.958.1 Detailed Description

Contains the parameters passed for indication about transport network registration change 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

#### Note

None

### 8.958.2 Field Documentation

8.958.2.1 `uint8_t unpack_sms_SLQSNWRegInfoCallback_ind_t::NWRegStat`

8.958.2.2 `swi_uint256_t unpack_sms_SLQSNWRegInfoCallback_ind_t::ParamPresenceMask`

## 8.959 `unpack_sms_SLQSSendAsyncSMS_t` Struct Reference

### Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.959.1 Detailed Description

This structure contains unpack SLQSSendAsyncSMS parameter.

#### Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.959.2 Field Documentation

8.959.2.1 `swi_uint256_t unpack_sms_SLQSSendAsyncSMS_t::ParamPresenceMask`

## 8.960 `unpack_sms_SLQSSetIndicationRegister_t` Struct Reference

## Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.960.1 Detailed Description

This structure contains unpack SLQSSetIndicationRegister parameter.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

### 8.960.2 Field Documentation

8.960.2.1 [swi\\_uint256\\_t unpack\\_sms\\_SLQSSetIndicationRegister\\_t::ParamPresenceMask](#)

## 8.961 unpack\_sms\_SLQSSetSmsBroadcastActivation\_t Struct Reference

## Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.961.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastActivation parameter.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

### 8.961.2 Field Documentation

8.961.2.1 [swi\\_uint256\\_t unpack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t::ParamPresenceMask](#)

## 8.962 unpack\_sms\_SLQSSetSmsBroadcastConfig\_t Struct Reference

## Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.962.1 Detailed Description

This structure contains unpack SLQSSetSmsBroadcastConfig parameters.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

### 8.962.2 Field Documentation

8.962.2.1 [swi\\_uint256\\_t unpack\\_sms\\_SLQSSetSmsBroadcastConfig\\_t::ParamPresenceMask](#)

## 8.963 [unpack\\_sms\\_SLQSSetSmsStorage\\_t](#) Struct Reference

### Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.963.1 Detailed Description

This structure contains unpack SLQSSetSmsStorage parameter.

#### Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.963.2 Field Documentation

8.963.2.1 [swi\\_uint256\\_t unpack\\_sms\\_SLQSSetSmsStorage\\_t::ParamPresenceMask](#)

## 8.964 [unpack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t](#) Struct Reference

### Data Fields

- [sms\\_maxStorageSizeResp](#) \* [pMaxStorageSizeResp](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.964.1 Detailed Description

Structure contain parameters that 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>pMaxStorageSizeResp</i>	<ul style="list-style-type: none"> <li>• Response parameters for SmsSLQSGetMaxStorageSize <ul style="list-style-type: none"> <li>– See <a href="#">sms_maxStorageSizeResp</a> for more information</li> </ul> </li> <li>• pMaxStorageSizeResp-&gt;maxStorageSize <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>1</b></li> </ul> </li> <li>• pMaxStorageSizeResp-&gt;freeSlots <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.964.2 Field Documentation

8.964.2.1 `swi_uint256_t` unpack\_sms\_SLQSSmsGetMaxStorageSize\_t::ParamPresenceMask

8.964.2.2 `sms_maxStorageSizeResp*` unpack\_sms\_SLQSSmsGetMaxStorageSize\_t::pMaxStorageSizeResp

## 8.965 unpack\_sms\_SLQSSmsGetMessageProtocol\_t Struct Reference

### Data Fields

- [sms\\_msgProtocolResp](#) \* [pMessageProtocol](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.965.1 Detailed Description

Structure contain Parameters that get the message protocol currently in use for the WMS client.

#### Parameters

<i>pMessage-Protocol</i>	<ul style="list-style-type: none"><li>• Pointer to <code>smsMsgprotocolResp</code><ul style="list-style-type: none"><li>– See <a href="#">sms_msgProtocolResp</a> for more information</li></ul></li><li>• Bit to check in <code>ParamPresenceMask</code> - 1</li></ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>

### 8.965.2 Field Documentation

8.965.2.1 `swi_uint256_t` unpack\_sms\_SLQSSmsGetMessageProtocol\_t::ParamPresenceMask

8.965.2.2 `sms_msgProtocolResp*` unpack\_sms\_SLQSSmsGetMessageProtocol\_t::pMessageProtocol

## 8.966 unpack\_sms\_SLQSSmsSetRoutes\_t Struct Reference

### Data Fields

- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.966.1 Detailed Description

This structure contains unpack SLQSSmsSetRoutes parameter.

#### Parameters

<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
---------------------------	--

### 8.966.2 Field Documentation

8.966.2.1 `swi_uint256_t` unpack\_sms\_SLQSSmsSetRoutes\_t::ParamPresenceMask

## 8.967 unpack\_sms\_SLQSSwiGetSMSStorage\_t Struct Reference

### Data Fields

- [uint32\\_t](#) \* [pSmsStorage](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.967.1 Detailed Description

Structure contain Parameters that return current SMS configuration that is applied to all incoming and outgoing messages.

#### Parameters

<i>pSmsStorage</i>	<ul style="list-style-type: none"> <li>• Values: <ul style="list-style-type: none"> <li>– 0x01 - device's permanent memory</li> <li>– 0x02 - UICC</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.967.2 Field Documentation

8.967.2.1 [swi\\_uint256\\_t](#) [unpack\\_sms\\_SLQSSwiGetSMSStorage\\_t::ParamPresenceMask](#)

8.967.2.2 [uint32\\_t](#)\* [unpack\\_sms\\_SLQSSwiGetSMSStorage\\_t::pSmsStorage](#)

## 8.968 unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t Struct Reference

### Data Fields

- [uint8\\_t](#) [regInd](#)
- [sms\\_transLayerInfo](#) \* [pTransLayerInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.968.1 Detailed Description

Contains the parameters passed for the indication about change in transport layer information

#### 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
---------------	---

<i>pTransLayerInfo</i>	<ul style="list-style-type: none"> <li>• Optional parameter</li> <li>• See <a href="#">sms_transLayerInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## Note

None

## 8.968.2 Field Documentation

8.968.2.1 swi\_uint256\_t unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t::ParamPresenceMask

8.968.2.2 sms\_transLayerInfo\* unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t::pTransLayerInfo

8.968.2.3 uint8\_t unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t::regInd

## 8.969 unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t Struct Reference

## Data Fields

- uint16\_t [sendStatus](#)
- uint16\_t [messageID](#)
- uint16\_t [causeCode](#)
- uint8\_t [errorClass](#)
- uint16\_t [RPCause](#)
- uint8\_t [TPCause](#)
- uint8\_t [msgDelFailureType](#)
- uint8\_t [msgDelFailureCause](#)
- uint8\_t [alphaDLen](#)
- uint8\_t \* [pAlphaID](#)
- uint32\_t [userData](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.969.1 Detailed Description

This structure contains SMS parameters for indication of RAW ASYNC SEND

## 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
-------------------	--

<i>messageID</i>	<ul style="list-style-type: none"> <li>• Unique ID assigned by WMS for non-retry messages.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>causeCode</i>	<ul style="list-style-type: none"> <li>• WMS cause code</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>RPCause</i>	<ul style="list-style-type: none"> <li>• GW RP cause</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>TPCause</i>	<ul style="list-style-type: none"> <li>• GW TP Cause</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>alphaIDLen</i>	<ul style="list-style-type: none"> <li>• Number of sets of the pAlphaID</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pAlphaID</i>	<ul style="list-style-type: none"> <li>• Alpha ID</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>userData</i>	<ul style="list-style-type: none"> <li>• Identifies the request associated with this indication.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>



## 8.969.2 Field Documentation

- 8.969.2.1 uint8\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::alphaDLen
- 8.969.2.2 uint16\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::causeCode
- 8.969.2.3 uint8\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::errorClass
- 8.969.2.4 uint16\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::messageID
- 8.969.2.5 uint8\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::msgDelFailureCause
- 8.969.2.6 uint8\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::msgDelFailureType
- 8.969.2.7 uint8\_t\* unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::pAlphaID
- 8.969.2.8 swi\_uint256\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::ParamPresenceMask
- 8.969.2.9 uint16\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::RPCause
- 8.969.2.10 uint16\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::sendStatus
- 8.969.2.11 uint8\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::TPCause
- 8.969.2.12 uint32\_t unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t::userData

## 8.970 unpack\_sms\_SLQSWmsMemoryFullCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [storageType](#)
- uint8\_t [messageMode](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.970.1 Detailed Description

This structure contains unpack SMS memory full callback indication.

#### 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> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.970.2 Field Documentation

8.970.2.1 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::messageMode`

8.970.2.2 `swi_uint256_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::ParamPresenceMask`

8.970.2.3 `uint8_t unpack_sms_SLQSWmsMemoryFullCallBack_ind_t::storageType`

## 8.971 `unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t` Struct Reference

### Data Fields

- `uint8_t numInstances`
- `sms_messageWaitingInfoContent msgWaitInfo` [0xFF]
- `swi_uint256_t ParamPresenceMask`

### 8.971.1 Detailed Description

This structure holds information related to message waiting information indication

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of sets of the elements in structure <code>sms_messageWaitingInfoContent</code></li> <li>• Bit to check in <code>ParamPresenceMask</code> - 1</li> </ul>
<i>msgWaitInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of <code>sms_messageWaitingInfoContent</code>. <ul style="list-style-type: none"> <li>– See <code>sms_messageWaitingInfoContent</code> for more information.</li> </ul> </li> <li>• Bit to check in <code>ParamPresenceMask</code> - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.971.2 Field Documentation

8.971.2.1 `sms_messageWaitingInfoContent unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::msgWaitInfo`[0xFF]

8.971.2.2 `uint8_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::numInstances`

8.971.2.3 `swi_uint256_t unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t::ParamPresenceMask`

## 8.972 `unpack_swiaudio_SLQSGetM2MAudioProfile_t` Struct Reference

### Data Fields

- `uint8_t Profile`
- `uint8_t EarMute`
- `uint8_t MicMute`
- `uint8_t Generator`
- `uint8_t Volume`
- `uint8_t CwtMute`

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.972.1 Detailed Description

This structure contains the SLQSGetM2MAudioProfile unpack 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>2</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></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> <li>• Bit to check in ParamPresenceMask - <b>4</b></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> <li>• Bit to check in ParamPresenceMask - <b>5</b></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> <li>• Bit to check in ParamPresenceMask - <b>6</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.972.2 Field Documentation

8.972.2.1 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::CwtMute`

8.972.2.2 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::EarMute`

8.972.2.3 `uint8_t unpack_swiaudio_SLQSGetM2MAudioProfile_t::Generator`

8.972.2.4 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::MicMute`

8.972.2.5 `swi_uint256_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::ParamPresenceMask`

8.972.2.6 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::Profile`

8.972.2.7 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioProfile_t::Volume`

## 8.973 `unpack_swiaudio_SLQSGetM2MAudioVolume_t` Struct Reference

### Data Fields

- `uint8_t` [Level](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.973.1 Detailed Description

This structure contains the `SLQSGetM2MAudioVolume` unpack 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> <li>• Bit to check in <code>ParamPresenceMask</code> - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.973.2 Field Documentation

8.973.2.1 `uint8_t` `unpack_swiaudio_SLQSGetM2MAudioVolume_t::Level`

8.973.2.2 `swi_uint256_t` `unpack_swiaudio_SLQSGetM2MAudioVolume_t::ParamPresenceMask`

## 8.974 `unpack_swiaudio_SLQSGetM2MAVMute_t` Struct Reference

### Data Fields

- `uint8_t` [EarMute](#)
- `uint8_t` [MicMute](#)
- `uint8_t` [CwtMute](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.974.1 Detailed Description

This structure contains the `SLQSGetM2MAVMute` unpack 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>2</b></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> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.974.2 Field Documentation

8.974.2.1 uint8\_t unpack\_swiaudio\_SLQSGetM2MAVMute\_t::CwtMute

8.974.2.2 uint8\_t unpack\_swiaudio\_SLQSGetM2MAVMute\_t::EarMute

8.974.2.3 uint8\_t unpack\_swiaudio\_SLQSGetM2MAVMute\_t::MicMute

8.974.2.4 swi\_uint256\_t unpack\_swiaudio\_SLQSGetM2MAVMute\_t::ParamPresenceMask

## 8.975 unpack\_swiaudio\_SLQSGetM2MSpkrGain\_t Struct Reference

## Data Fields

- uint16\_t [Value](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.975.1 Detailed Description

This structure contains the SLQSGetM2MSpkrGain unpack 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> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.975.2 Field Documentation

8.975.2.1 `swi_uint256_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::ParamPresenceMask`

8.975.2.2 `uint16_t unpack_swiaudio_SLQSGetM2MSpkrGain_t::Value`

## 8.976 `unpack_swiaavs_SLQSAVMSEventReportInd_t` Struct Reference

### Data Fields

- [UnpackSwiAvmsEventReportBinaryUpdateSessionInfo](#) \* `pBinaryUpdateSessionInfo`
- [UnpackSwiAvmsEventReportConfig](#) \* `pConfig`
- [UnpackSwiAvmsEventReportNotification](#) \* `pNotification`
- [UnpackSwiAvmsEventReportPackageID](#) \* `pPackageID`
- [UnpackSwiAvmsEventReportConnectionRequest](#) \* `pConnectionRequest`
- [UnpackSwiAvmsEventReportWAMSPParamChange](#) \* `pWAMSPParaChanged`
- [UnpackSwiAvmsEventReportRegStatus](#) \* `pRegStatus`
- [UnpackSwiAvmsEventReportDataSessionStatus](#) \* `pDataSessionStatus`
- [UnpackSwiAvmsEventReportSessionType](#) \* `pSessionType`
- [UnpackSwiAvmsEventReportHTTPStatus](#) \* `pHTTPStatus`
- `uint32_t resultCode`
- `swi_uint256_t ParamPresenceMask`

### 8.976.1 Detailed Description

Structure that contains the session type response for AVMS event report indication command

#### Parameters

<i>pBinaryUpdateSessionInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportBinaryUpdateSessionInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pConfig</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportConfig</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pNotification</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportNotification</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pPackageID</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportPackageID</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pConnectionRequest</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportConnectionRequest</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pWAMSPParaChanged</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportWAMSPParamChange</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>

<i>pRegStatus[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportRegStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pDataSessionStatus[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportDataSessionStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pSessionType[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportSessionType</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pHTTPStatus[OUT]</i>	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportHTTPStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>resultcode</i>	- Tlv Result Code.

## 8.976.2 Field Documentation

8.976.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSEventReportInd_t::ParamPresenceMask`

8.976.2.2 `UnpackSwiAvmsEventReportBinaryUpdateSessionInfo* unpack_swiavms_SLQSAVMSEventReportInd_t::pBinaryUpdateSessionInfo`

8.976.2.3 `UnpackSwiAvmsEventReportConfig* unpack_swiavms_SLQSAVMSEventReportInd_t::pConfig`

8.976.2.4 `UnpackSwiAvmsEventReportConnectionRequest* unpack_swiavms_SLQSAVMSEventReportInd_t::pConnectionRequest`

8.976.2.5 `UnpackSwiAvmsEventReportDataSessionStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pDataSessionStatus`

8.976.2.6 `UnpackSwiAvmsEventReportHTTPStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pHTTPStatus`

8.976.2.7 `UnpackSwiAvmsEventReportNotification* unpack_swiavms_SLQSAVMSEventReportInd_t::pNotification`

8.976.2.8 `UnpackSwiAvmsEventReportPackageID* unpack_swiavms_SLQSAVMSEventReportInd_t::pPackageID`

8.976.2.9 `UnpackSwiAvmsEventReportRegStatus* unpack_swiavms_SLQSAVMSEventReportInd_t::pRegStatus`

8.976.2.10 `UnpackSwiAvmsEventReportSessionType* unpack_swiavms_SLQSAVMSEventReportInd_t::pSessionType`

8.976.2.11 `UnpackSwiAvmsEventReportWAMSParamChange* unpack_swiavms_SLQSAVMSEventReportInd_t::pWAMSParaChanged`

8.976.2.12 `uint32_t unpack_swiavms_SLQSAVMSEventReportInd_t::resultcode`

## 8.977 unpack\_swiavms\_SLQSAVMGetSettings\_t Struct Reference

### Data Fields

- `uint32_t OMADMEnabled`

- uint8\_t [AutoConnect](#)
- uint8\_t [AutoReboot](#)
- uint8\_t [FwPromptdownload](#)
- uint8\_t [FwPromptUpdate](#)
- uint8\_t [FwAutoSDM](#)
- uint32\_t \* [pPollingTimer](#)
- [PackSwiAVMSSettingsConnectionRetryTimers](#) \* [pConnectionRetryTimers](#)
- [PackSwiAVMSSettingsAPNInfo](#) \* [pAPNInfo](#)
- uint8\_t \* [pNotificationStore](#)
- [PackSwiAVMSSettingsPeriodsInfo](#) \* [pPeroidsInfo](#)
- uint32\_t [resultcode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.977.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

#### Parameters

<i>OMADMEEnabled</i>	- OMA DM Enabled. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>
----------------------	---

- Bit to check in ParamPresenceMask - **1**

#### Parameters

<i>AutoConnect</i>	- Auto Connect. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>
--------------------	---

- Bit to check in ParamPresenceMask - **6**

#### Parameters

<i>AutoReboot</i>	- Auto Reboot. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>
-------------------	--

- Bit to check in ParamPresenceMask - **21**

#### Parameters

<i>Fw-Autodownload</i>	- Firmware auto download. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul> – Bit to check in ParamPresenceMask - <b>3</b>
<i>FwAutoUpdate</i>	- Firmware auto update. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>



- Bit to check in ParamPresenceMask - 4

## Parameters

<i>FwAutoSDM</i>	- OMA Automatic UI Alert Response. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>
------------------	--

- Bit to check in ParamPresenceMask - 5

## Parameters

<i>pPollingTimer</i>	- Polling timer to connect to AVMS server. <ul style="list-style-type: none"> <li>• 0-525600 (min)</li> <li>• 0:disabled</li> </ul>
----------------------	---

- Bit to check in ParamPresenceMask - 16

## Parameters

<i>pConnection-RetryTimers</i>	- Connection Retry timers. <ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAVMSSettingsConnectionRetryTimers</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul>
<i>pAPNInfo</i>	- APN Information. <ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAVMSSettingsAPNInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 18</li> </ul>
<i>pNotification-Store</i>	- Notification Storing When Disabled or Offline. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>

- Bit to check in ParamPresenceMask - 19

## Parameters

<i>pPeroidsInfo</i>	- Min and Max Period of an Observation. <ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAVMSSettingsPeriodsInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 20</li> </ul>
<i>resultcode</i>	- Tlv Result Code.

## 8.977.2 Field Documentation

8.977.2.1 uint8\_t unpack\_swiaavms\_SLQSAVMSGetSettings\_t::AutoConnect

8.977.2.2 uint8\_t unpack\_swiaavms\_SLQSAVMSGetSettings\_t::AutoReboot

8.977.2.3 uint8\_t unpack\_swiaavms\_SLQSAVMSGetSettings\_t::FwAutoSDM

- 8.977.2.4 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptdownload`
- 8.977.2.5 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_t::FwPromptUpdate`
- 8.977.2.6 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::OMADMEEnabled`
- 8.977.2.7 `PackSwiAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pAPNInfo`
- 8.977.2.8 `swi_uint256_t unpack_swiavms_SLQSAVMSGetSettings_t::ParamPresenceMask`
- 8.977.2.9 `PackSwiAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSGetSettings_t::pConnection-RetryTimers`
- 8.977.2.10 `uint8_t* unpack_swiavms_SLQSAVMSGetSettings_t::pNotificationStore`
- 8.977.2.11 `PackSwiAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSGetSettings_t::pPeroidsInfo`
- 8.977.2.12 `uint32_t* unpack_swiavms_SLQSAVMSGetSettings_t::pPollingTimer`
- 8.977.2.13 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_t::resultcode`

## 8.978 `unpack_swiavms_SLQSAVMSGetSettings_v2_t` Struct Reference

### Data Fields

- `uint32_t` [OMADMEEnabled](#)
- `uint8_t` [AutoConnect](#)
- `uint8_t` [FwPromptdownload](#)
- `uint8_t` [FwPromptUpdate](#)
- `uint8_t` [FwAutoSDM](#)
- `uint32_t *` [pPollingTimer](#)
- [PackSwiAVMSSettingsConnectionRetryTimers](#) \* [pConnectionRetryTimers](#)
- [PackSwiAVMSSettingsAPNInfo](#) \* [pAPNInfo](#)
- `uint8_t *` [pNotificationStore](#)
- [PackSwiAVMSSettingsPeriodsInfo](#) \* [pPeroidsInfo](#)
- `uint8_t *` [pAutoReboot](#)
- `uint32_t` [resultcode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.978.1 Detailed Description

Structure that contains the response for AVMS Get Settings command It maps with SLQSAVMSGetSettings\_v2 (For AVC2 service)

#### Parameters

<i>OMADMEEnabled</i>	- OMA DM Enabled. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul>
----------------------	---

- Bit to check in ParamPresenceMask - 1

## Parameters

<i>AutoConnect</i>	<ul style="list-style-type: none"> <li>- Auto Connect. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul> </li> </ul>
--------------------	--

- Bit to check in ParamPresenceMask - 6

## Parameters

<i>Fw-Promptdownload</i>	<ul style="list-style-type: none"> <li>- Firmware Prompt download. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled. <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - 3</li> </ul> </li> </ul> </li> </ul>
<i>FwPrompt-Update</i>	<ul style="list-style-type: none"> <li>- Firmware Prompt update. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul> </li> </ul>

- Bit to check in ParamPresenceMask - 4

## Parameters

<i>FwAutoSDM</i>	<ul style="list-style-type: none"> <li>- OMA Automatic UI Alert Response. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.</li> </ul> </li> </ul>
------------------	---

- Bit to check in ParamPresenceMask - 5

## Parameters

<i>pPollingTimer[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>- Polling timer to connect to AVMS server. <ul style="list-style-type: none"> <li>• 0-525600 (min)</li> <li>• 0:disabled <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - 16</li> </ul> </li> </ul> </li> </ul>
<i>pConnection-RetryTimers[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>- Connection Retry timers. <ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAVMSSettingsConnectionRetryTimers</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul> </li> </ul>
<i>pAPNInfo[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>- APN Information. <ul style="list-style-type: none"> <li>• See <a href="#">PackSwiAVMSSettingsAPNInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 18</li> </ul> </li> </ul>
<i>pNotification-Store[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>- Notification Storing When Disabled or Offline. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled. <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - 19</li> </ul> </li> </ul> </li> </ul>

<i>pPeriodsInfo</i> [OPTIONAL]	- Min and Max Period of an Observation. <ul style="list-style-type: none"> <li>• See <a href="#">PackSviAVMSSettingsPeriodsInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pAutoReboot</i> [OPTIONAL]	- Auto Reboot. <ul style="list-style-type: none"> <li>• 0 - Disabled.</li> <li>• 1 - Enabled.             <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>21</b></li> </ul> </li> </ul>
<i>resultcode</i>	- Tlv Result Code.

## 8.978.2 Field Documentation

8.978.2.1 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::AutoConnect`

8.978.2.2 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwAutoSDM`

8.978.2.3 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwPromptdownload`

8.978.2.4 `uint8_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::FwPromptUpdate`

8.978.2.5 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::OMADMEEnabled`

8.978.2.6 `PackSviAVMSSettingsAPNInfo* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pAPNInfo`

8.978.2.7 `swi_uint256_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::ParamPresenceMask`

8.978.2.8 `uint8_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pAutoReboot`

8.978.2.9 `PackSviAVMSSettingsConnectionRetryTimers* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pConnectionRetryTimers`

8.978.2.10 `uint8_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pNotificationStore`

8.978.2.11 `PackSviAVMSSettingsPeriodsInfo* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pPeriodsInfo`

8.978.2.12 `uint32_t* unpack_swiavms_SLQSAVMSGetSettings_v2_t::pPollingTimer`

8.978.2.13 `uint32_t unpack_swiavms_SLQSAVMSGetSettings_v2_t::resultcode`

## 8.979 unpack\_swiavms\_SLQSAVMSSendSelection\_t Struct Reference

### Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

### 8.979.1 Detailed Description

Structure that contains the session type response for AVMS Selection command

## Parameters

<i>resultcode</i>	- Tlv Result Code. • Bit to check in ParamPresenceMask - 2
-------------------	---

## 8.979.2 Field Documentation

8.979.2.1 swi\_uint256\_t unpack\_swiaavms\_SLQSAVMSSendSelection\_t::ParamPresenceMask

8.979.2.2 uint32\_t unpack\_swiaavms\_SLQSAVMSSendSelection\_t::resultcode

## 8.980 unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t Struct Reference

## Data Fields

- [UnpackSwiAvmsEventReportBinaryUpdateSessionInfo](#) \* pBinaryUpdateSessionInfo
- [UnpackSwiAvmsEventReportConfig](#) \* pConfig
- [UnpackSwiAvmsEventReportNotification](#) \* pNotification
- [UnpackSwiAvmsEventReportPackageID](#) \* pPackageID
- uint32\_t resultcode
- swi\_uint256\_t ParamPresenceMask

## 8.980.1 Detailed Description

Structure that contains the session type response for AVMS get session info command

## Parameters

<i>pBinaryUpdateSessionInfo</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportBinaryUpdateSessionInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>pConfig</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportConfig</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul>
<i>pNotification</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportNotification</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 18</li> </ul>
<i>pPackageID</i> [OUT]	<ul style="list-style-type: none"> <li>• See <a href="#">UnpackSwiAvmsEventReportPackageID</a> for more information</li> <li>• Bit to check in ParamPresenceMask - 19</li> </ul>
<i>resultcode</i>	- Tlv Result Code.

## 8.980.2 Field Documentation

8.980.2.1 swi\_uint256\_t unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t::ParamPresenceMask

8.980.2.2 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo\* unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t::pBinaryUpdateSessionInfo

8.980.2.3 **UnpackSwiAvmsEventReportConfig\*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pConfig`

8.980.2.4 **UnpackSwiAvmsEventReportNotification\*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pNotification`

8.980.2.5 **UnpackSwiAvmsEventReportPackageID\*** `unpack_swiavms_SLQSAVMSSessionGetInfo_t::pPackageID`

8.980.2.6 `uint32_t unpack_swiavms_SLQSAVMSSessionGetInfo_t::resultcode`

## 8.981 `unpack_swiavms_SLQSAvmsSetEventReport_t` Struct Reference

### Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

### 8.981.1 Detailed Description

Structure that contains the session type response for AVMS set event report command

#### Parameters

<i>resultcode</i>	- Tlv Result Code. <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
-------------------	--

### 8.981.2 Field Documentation

8.981.2.1 `swi_uint256_t unpack_swiavms_SLQSAvmsSetEventReport_t::ParamPresenceMask`

8.981.2.2 `uint32_t unpack_swiavms_SLQSAvmsSetEventReport_t::resultcode`

## 8.982 `unpack_swiavms_SLQSAVMSSetSettings_t` Struct Reference

### Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

### 8.982.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

#### Parameters

<i>resultcode</i>	- Tlv Result Code. <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
-------------------	--

### 8.982.2 Field Documentation

8.982.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSSetSettings_t::ParamPresenceMask`

8.982.2.2 uint32\_t unpack\_swiavms\_SLQSAVMSSetSettings\_t::resultcode

## 8.983 unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t Struct Reference

### Data Fields

- uint32\_t [resultcode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

#### 8.983.1 Detailed Description

Structure that contains the session type response for AVMS Set Settings command

##### Parameters

<i>resultcode</i>	- Tlv Result Code. <ul style="list-style-type: none"><li>Bit to check in ParamPresenceMask - <b>2</b></li></ul>
-------------------	---

#### 8.983.2 Field Documentation

8.983.2.1 [swi\\_uint256\\_t](#) unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t::ParamPresenceMask

8.983.2.2 uint32\_t unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t::resultcode

## 8.984 unpack\_swiavms\_SLQSAVMSStartSession\_t Struct Reference

### Data Fields

- uint32\_t [sessionResponse](#)
- uint32\_t [resultcode](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

#### 8.984.1 Detailed Description

Structure that contains the session type response for AVMS start session command

##### Parameters

<i>Session-Response</i>	- OMA-DM availability CHECK for the requested item. <ul style="list-style-type: none"><li>0x00000001 - Available.</li><li>0x00000002 - Not Available.</li><li>0x00000003 - Check Timed Out.</li><li>Bit to check in ParamPresenceMask - <b>1</b></li></ul>
<i>resultcode</i>	- Tlv Result Code.

#### 8.984.2 Field Documentation

8.984.2.1 [swi\\_uint256\\_t](#) unpack\_swiavms\_SLQSAVMSStartSession\_t::ParamPresenceMask

8.984.2.2 `uint32_t unpack_swiavms_SLQSAVMSStartSession_t::resultcode`

8.984.2.3 `uint32_t unpack_swiavms_SLQSAVMSStartSession_t::sessionResponse`

## 8.985 `unpack_swiavms_SLQSAVMSStopSession_t` Struct Reference

### Data Fields

- `uint32_t resultcode`
- `swi_uint256_t ParamPresenceMask`

### 8.985.1 Detailed Description

Structure that contains the session type response for AVMS Stop session command

#### Parameters

<i>resultcode</i>	<ul style="list-style-type: none"> <li>- Tlv Result Code.</li> <li>• Bit to check in ParamPresenceMask - <b>2</b></li> </ul>
-------------------	--

### 8.985.2 Field Documentation

8.985.2.1 `swi_uint256_t unpack_swiavms_SLQSAVMSStopSession_t::ParamPresenceMask`

8.985.2.2 `uint32_t unpack_swiavms_SLQSAVMSStopSession_t::resultcode`

## 8.986 `unpack_swidms_SLQSSwiDmsGetHWWatchdog_t` Struct Reference

### Data Fields

- `swidms_SwiDmsGetHWWatchdog * pHWWatchdog`
- `swi_uint256_t ParamPresenceMask`

### 8.986.1 Detailed Description

This structure contains the get hw watchdog response

#### Parameters

<i>pHWWatchdog</i>	<ul style="list-style-type: none"> <li>• See <code>swidms_SwiDmsGetHWWatchdog</code> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.986.2 Field Documentation

8.986.2.1 `swi_uint256_t unpack_swidms_SLQSSwiDmsGetHWWatchdog_t::ParamPresenceMask`



8.986.2.2 swidms\_SwiDmsGetHWWatchdog\* unpack\_swidms\_SLQSSwiDmsGetHWWatchdog\_t::pHWWatchdog

## 8.987 unpack\_swidms\_SLQSSwiDmsGetMTU\_t Struct Reference

### Data Fields

- [swidms\\_mtuSize3gppTlv](#) \* [pMTUSize3gpp](#)
- [swidms\\_hrpdMTUSizeTlv](#) \* [pHrpdMTUSize](#)
- [swidms\\_ehrpdMTUSizeTlv](#) \* [pEhrpdMTUSize](#)
- [swidms\\_usbMTUSizeTlv](#) \* [pUsbMTUSize](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.987.1 Detailed Description

This structure contains the Get MTU Response parameter.

#### Parameters

<i>pMTUSize3gpp</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">swidms_mtuSize3gppTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHrpdMTUSize</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">swidms_hrpdMTUSizeTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pEhrpdMTUSize</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">swidms_ehrpdMTUSizeTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pUsbMTUSize</i>	[Optional] <ul style="list-style-type: none"> <li>• See <a href="#">swidms_usbMTUSizeTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.987.2 Field Documentation

8.987.2.1 swi\_uint256\_t unpack\_swidms\_SLQSSwiDmsGetMTU\_t::ParamPresenceMask

8.987.2.2 swidms\_ehrpdMTUSizeTlv\* unpack\_swidms\_SLQSSwiDmsGetMTU\_t::pEhrpdMTUSize

8.987.2.3 swidms\_hrpdMTUSizeTlv\* unpack\_swidms\_SLQSSwiDmsGetMTU\_t::pHrpdMTUSize

8.987.2.4 swidms\_mtuSize3gppTlv\* unpack\_swidms\_SLQSSwiDmsGetMTU\_t::pMTUSize3gpp

8.987.2.5 swidms\_usbMTUSizeTlv\* unpack\_swidms\_SLQSSwiDmsGetMTU\_t::pUsbMTUSize

## 8.988 unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t Struct Reference

## Data Fields

- uint8\_t [secureBootEnabled](#)
- uint8\_t [memoryDumpAllowed](#)
- uint8\_t [jtagAccessAllowed](#)
- uint16\_t [TlvResult](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.988.1 Detailed Description

This structure contains secure boot config and other capabilities

#### Parameters

<i>secureBoot-Enabled</i>	<ul style="list-style-type: none"> <li>• secure boot enabled or disabled 0- disabled 1- enabled</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>memoryDump-Allowed</i>	<ul style="list-style-type: none"> <li>• memory dump allowed or disallowed 0- disallowed 1- allowed</li> <li>• Bit to check in ParamPresenceMask - <b>3</b></li> </ul>
<i>jtagAccess-Allowed</i>	<ul style="list-style-type: none"> <li>• Jtag access allowed or disallowed 0- disallowed 1- allowed</li> <li>• Bit to check in ParamPresenceMask - <b>4</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> <li>• Bit to check in ParamPresenceMask - <b>2</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.988.2 Field Documentation

8.988.2.1 uint8\_t unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t::jtagAccessAllowed

8.988.2.2 uint8\_t unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t::memoryDumpAllowed

8.988.2.3 swi\_uint256\_t unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t::ParamPresenceMask

8.988.2.4 uint8\_t unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t::secureBootEnabled

8.988.2.5 uint16\_t unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t::TlvResult

## 8.989 unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t Struct Reference

### Data Fields

- swidms\_interfaceCfgTlv \* [pInterfaceCfg](#)
- swidms\_supportedIntBitmaskTlv \* [pSupportedBitmasks](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.989.1 Detailed Description

This structure contains the get usb composition response

#### Parameters

<i>pInterfaceCfg</i>	<ul style="list-style-type: none"> <li>See <a href="#">swidms_ifaceCfgTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pSupported-Bitmaps</i>	[Optional] <ul style="list-style-type: none"> <li>See <a href="#">swidms_supportedIntBitmaskTlv</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.989.2 Field Documentation

8.989.2.1 `swi_uint256_t` unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t::ParamPresenceMask

8.989.2.2 `swidms_ifaceCfgTlv*` unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t::pInterfaceCfg

8.989.2.3 `swidms_supportedIntBitmaskTlv*` unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t::pSupportedBitmaps

## 8.990 unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t Struct Reference

#### Data Fields

- `uint8_t` [usbNetNum](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.990.1 Detailed Description

This structure contains usb net numbers to get from remote endpoint for QMAP configuration

#### Parameters

<i>usbNetNum</i>	<ul style="list-style-type: none"> <li>value of usb net numbers on the device</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.990.2 Field Documentation

8.990.2.1 `swi_uint256_t` unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t::ParamPresenceMask

8.990.2.2 `uint8_t` unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t::usbNetNum

## 8.991 `unpack_swidms_SLQSSwiDmsSetHWWatchdog_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.991.1 Detailed Description

This structure is used to store [unpack\\_swidms\\_SLQSSwiDmsSetHWWatchdog\\_t](#) parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.991.2 Field Documentation

8.991.2.1 `swi_uint256_t` `unpack_swidms_SLQSSwiDmsSetHWWatchdog_t::ParamPresenceMask`

8.991.2.2 `uint16_t` `unpack_swidms_SLQSSwiDmsSetHWWatchdog_t::Tlvresult`

## 8.992 `unpack_swidms_SLQSSwiDmsSetMTU_t` Struct Reference

### Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.992.1 Detailed Description

This structure is used to store [unpack\\_swidms\\_SLQSSwiDmsSetMTU\\_t](#) parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.992.2 Field Documentation

8.992.2.1 `swi_uint256_t` `unpack_swidms_SLQSSwiDmsSetMTU_t::ParamPresenceMask`

8.992.2.2 uint16\_t unpack\_swidms\_SLQSSwiDmsSetMTU\_t::Tlvresult

## 8.993 unpack\_swidms\_SLQSSwiDmsSetUsbComp\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.993.1 Detailed Description

This structure is used to store [unpack\\_swidms\\_SLQSSwiDmsSetUsbComp\\_t](#) parameters.

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• unpack Tlv result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.993.2 Field Documentation

8.993.2.1 [swi\\_uint256\\_t](#) unpack\_swidms\_SLQSSwiDmsSetUsbComp\_t::ParamPresenceMask

8.993.2.2 uint16\_t unpack\_swidms\_SLQSSwiDmsSetUsbComp\_t::Tlvresult

## 8.994 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](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.994.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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>

## 8.994.2 Field Documentation

- 8.994.2.1 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::fix_rate`
- 8.994.2.2 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_rate_reported`
- 8.994.2.3 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::fix_type`
- 8.994.2.4 `int unpack_swiloc_SwiLocGetAutoStart_t::fix_type_reported`
- 8.994.2.5 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::function`
- 8.994.2.6 `int unpack_swiloc_SwiLocGetAutoStart_t::function_reported`
- 8.994.2.7 `uint32_t unpack_swiloc_SwiLocGetAutoStart_t::max_dist`
- 8.994.2.8 `int unpack_swiloc_SwiLocGetAutoStart_t::max_dist_reported`
- 8.994.2.9 `uint8_t unpack_swiloc_SwiLocGetAutoStart_t::max_time`
- 8.994.2.10 `int unpack_swiloc_SwiLocGetAutoStart_t::max_time_reported`
- 8.994.2.11 `swi_uint256_t unpack_swiloc_SwiLocGetAutoStart_t::ParamPresenceMask`

## 8.995 unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t Struct Reference

### Data Fields

- `uint32_t eventType`
- `unpack_omaDmFotaTlv_t SessionInfoFota`
- `unpack_omaDmConfigTlv_t SessionInfoConfig`
- `unpack_omaDmNotificationsTlv_t SessionInfoNotification`
- `swi_uint256_t ParamPresenceMask`

### 8.995.1 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> <li>• 0xff - indication missing event information.</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.995.2 Field Documentation

8.995.2.1 `uint32_t unpack_swioama_SLQSOMADMAAlertCallback_ind_t::eventType`

8.995.2.2 `swi_uint256_t unpack_swioama_SLQSOMADMAAlertCallback_ind_t::ParamPresenceMask`

8.995.2.3 `unpack_omaDmConfigTlv_t unpack_swioama_SLQSOMADMAAlertCallback_ind_t::SessionInfoConfig`

8.995.2.4 `unpack_omaDmFotaTlv_t unpack_swioama_SLQSOMADMAAlertCallback_ind_t::SessionInfoFota`

8.995.2.5 `unpack_omaDmNotificationsTlv_t unpack_swioama_SLQSOMADMAAlertCallback_ind_t::SessionInfoNotification`

## 8.996 unpack\_swioama\_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`
- `swi_uint256_t ParamPresenceMask`

### 8.996.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>SourceLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Vendor Name String in Bytes.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">LITEQMI_MAX_SWIOMA_STR_LEN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>PkgNameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in Bytes.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">LITEQMI_MAX_SWIOMA_STR_LEN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>PkgDescLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">LITEQMI_MAX_SWIOMA_STR_LEN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>

<i>DateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Description String in Bytes.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">LITEQMI_MAX_SWIOMA_STR_LEN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>TimeLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Time String in Bytes.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">LITEQMI_MAX_SWIOMA_STR_LEN</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.996.2 Field Documentation

### 8.996.2.1 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfo\_t::Date[255]

- 8.996.2.2 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::DateLength
- 8.996.2.3 swi\_uint256\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::ParamPresenceMask
- 8.996.2.4 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::PkgDescLength
- 8.996.2.5 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::PkgDescription[255]
- 8.996.2.6 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::PkgName[255]
- 8.996.2.7 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::PkgNameLength
- 8.996.2.8 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::RetryCount
- 8.996.2.9 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::SessionState
- 8.996.2.10 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::SessionType
- 8.996.2.11 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::Severity
- 8.996.2.12 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::Source[255]
- 8.996.2.13 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::SourceLength
- 8.996.2.14 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::Status
- 8.996.2.15 uint8\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::Time[255]
- 8.996.2.16 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::TimeLength
- 8.996.2.17 uint16\_t unpack\_swioama\_SLQSOMADMGetSessionInfo\_t::UpdateCompleteStatus

## 8.997 unpack\_swioama\_SLQSOMADMGetSessionInfoExt\_t Struct Reference

### Data Fields

- uint8\_t [status](#)
- uint8\_t [sessionState](#)
- uint8\_t [hfaStatus](#)
- uint16\_t [hfaMaxRetry](#)
- uint16\_t [hfaRetryInterval](#)
- uint16\_t [hfaRetryIndex](#)
- uint8\_t [fumoState](#)
- uint32\_t [fumoResultCode](#)
- uint16\_t [pkgVendorNameLength](#)
- uint8\_t [pkgVendorName](#) [512]
- uint32\_t [pkgSize](#)
- uint16\_t [pkgVersionNameLength](#)
- uint8\_t [pkgVersionName](#) [512]
- uint16\_t [pkgNameLength](#)
- uint8\_t [pkgName](#) [512]
- uint16\_t [pkgDescLength](#)
- uint8\_t [pkgDesc](#) [512]
- uint16\_t [pkgDateLength](#)
- uint8\_t [pkgDate](#) [512]

- uint16\_t pkgInstallTimeLength
- uint8\_t pkgInstallTime [512]
- swi\_uint256\_t ParamPresenceMask

### 8.997.1 Detailed Description

Structure containing info for OMADM session Bit to check in ParamPresenceMask - 16

#### Parameters

<i>status</i>	<ul style="list-style-type: none"> <li>• 1 Byte parameter indicating status of OMADM initialization 0 – OMA module initiation not completed. 1 – OMA module initiation completed but no UI is registered. 128 – OMA initiation completed. 255 – OMA initiation error.</li> </ul>
<i>sessionState</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating OMA session state 0 - OMA module in CIDC session. 1 - OMA module in CIFUMO session 2 - OMA module in device initiated FUMO session 3 - OMA module in FUMO resume session 4 - OMA module is sending FUMO report 5 - OMA module in device initiated DM session 6 - OMA module in HFA DC session 7 - OMA module in CI PRL session 8 - OMA module in device initiated PRL session 9 - OMA module in HFA PRL session 10 - OMA module in HFA FUMO session 11 - OMA module in NI PRL session 12 - OMA is storing configuration after DC/PRL session 13 - OMA module is blocked by UI 14 - there is a pending session 15 - OMA module in idle state</li> </ul>
<i>hfaStatus</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating session state 0 – HFA completed 1 – HFA not start 2 – HFA DC completed, but PRL is not completed 3 – HFA PRL completed, but FUMO is not completed</li> </ul>
<i>hfaMaxRetry</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating max retry for HFA 0 – HFA completed</li> </ul>
<i>hfaRetryInterval</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating retry interval in seconds for HFA 0 – HFA completed</li> </ul>
<i>hfaRetryIndex</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating number of retry currently ongoing 0 – no retrying 0xFFFF- Invalid value</li> </ul>
<i>fumoState</i>	<ul style="list-style-type: none"> <li>• 1 byte parameter indicating fumo state 0 – FUMO state READY 1 – FUMO state DISCOVERY 2 – FUMO state DISCOCONF 3 – FUMO state DOWNLOAD 4 – FUMO state RECEIVED 5 – FUMO state CONFIRM 6 – FUMO state CONFIRMED 7 – FUMO state UPDATE 8 – FUMO state UPDATING 9 – FUMO state UNUSABLE 10 – FUMO state REJECTED 11 – FUMO state REJECTCONF 12 – FUMO state FAILED 13 – FUMO state DENIED 14 – FUMO state UPDATED 15 – FUMO state FINCONF 16 – FUMO state FINISHED</li> </ul>

<i>fumoResultCode</i>	<ul style="list-style-type: none"> <li>• 4 byte parameter indicating FUMO update image installation result. 200 - Request succeeded 250 – the start of vendor specified success result code 299 – the end of vendor specified success result code 400 - Management client error 401 - User rejected operation 402 - Corrupted update package 403 - Wrong package for device 404 - Invalid package signature 405 - Update package not acceptable 406 - DL auth failure 407 - DL download timeout 408 - Unsupported operation 409 - Err not defined by other code 410 - Firmware update failed 411 - Malformed or bad DL URL 412 - DL server unavailable 450 - vendor specified client error start 499 - vendor specified client error end 500 - DL server error 501 - DL fails due to out of memory 502 - Update fails, out of memory 503 - DL fails due to net issues 550 - vendor defined DL server error start 599 - vendor defined DL server error end 0xFFFFFFFF – invalid value.</li> </ul>
<i>pkgVendor-NameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of package vendor String in WORDs</li> </ul>
<i>pkgVendorName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in UCS2</li> <li>• size in bytes is 2*pkgVendorNameLength</li> </ul>
<i>pkgSize</i>	<ul style="list-style-type: none"> <li>• 4 byte parameter indicating pkg size</li> </ul>
<i>pkgVersion-NameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package version Name String in WORDs</li> </ul>
<i>pkgVersion-Name</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package version Name in UCS2</li> <li>• size in bytes is 2*pkgVersionNameLength</li> </ul>
<i>pkgNameLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package Name String in WORDs</li> </ul>
<i>pkgName</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package Name in UCS2</li> <li>• size in bytes is 2*pkgNameLength</li> </ul>
<i>pkgDescLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package description String in WORDs</li> </ul>
<i>pkgDesc</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package description in UCS2</li> <li>• size in bytes is 2*pkgDescLength</li> </ul>
<i>pkgDateLength</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package date String in WORDs</li> </ul>
<i>pkgDate</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package date in UCS2</li> <li>• size in bytes is 2*pkgDateLength</li> </ul>
<i>pkgInstallTime-Length</i>	<ul style="list-style-type: none"> <li>• 2 byte parameter indicating Length of Package install time String in WORDs</li> </ul>

<i>pkgInstallTime</i>	<ul style="list-style-type: none"> <li>• Variable length parameter indicating Package install time in UCS2</li> <li>• size in bytes is 2*pkgInstallTimeLength</li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.997.2 Field Documentation

8.997.2.1 uint32\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::fumoResultCode

8.997.2.2 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::fumoState

8.997.2.3 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::hfaMaxRetry

8.997.2.4 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::hfaRetryIndex

8.997.2.5 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::hfaRetryInterval

8.997.2.6 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::hfaStatus

8.997.2.7 swi\_uint256\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::ParamPresenceMask

8.997.2.8 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgDate[512]

8.997.2.9 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgDateLength

8.997.2.10 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgDesc[512]

8.997.2.11 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgDescLength

8.997.2.12 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgInstallTime[512]

8.997.2.13 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgInstallTimeLength

8.997.2.14 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgName[512]

8.997.2.15 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgNameLength

8.997.2.16 uint32\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgSize

8.997.2.17 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgVendorName[512]

8.997.2.18 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgVendorNameLength

8.997.2.19 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgVersionName[512]

8.997.2.20 uint16\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::pkgVersionNameLength

8.997.2.21 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::sessionState

8.997.2.22 uint8\_t unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t::status

## 8.998 unpack\_swima\_SLQSOMADMGetSettings\_t Struct Reference

### Data Fields

- uint32\_t [OMADMEEnabled](#)
- uint8\_t [FOTAdownload](#)
- uint8\_t [FOTAUpdate](#)
- uint8\_t [Autosdm](#)
- uint8\_t [FwAutoCheck](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.998.1 Detailed Description

Structure containing the OMA DM settings retrieved from the device

#### Parameters

<i>OMADMEEnabled</i>	[out] <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>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>FOTAdownload</i>	[out] <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>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>FOTAUpdate</i>	[out] <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>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>Autosdm</i>	[out] <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>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>FwAutoCheck</i>	[out] <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>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.998.2 Field Documentation

8.998.2.1 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::Autosdm

8.998.2.2 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FOTAdownload

8.998.2.3 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FOTAUpdate

8.998.2.4 uint8\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::FwAutoCheck

8.998.2.5 uint32\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::OMADMEEnabled

8.998.2.6 swi\_uint256\_t unpack\_swioama\_SLQSOMADMGetSettings\_t::ParamPresenceMask

## 8.999 unpack\_swioama\_SLQSOMADMStartSession\_t Struct Reference

### Data Fields

- uint32\_t [FwAvailability](#)
- swi\_uint256\_t [ParamPresenceMask](#)

### 8.999.1 Detailed Description

Structure that contains the responses for OMA start session command

#### Parameters

<i>FwAvailability</i>	[out] <ul style="list-style-type: none"> <li>OMA-DM CHECK FW Available</li> <li>Values             <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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>



### 8.999.2 Field Documentation

8.999.2.1 uint32\_t unpack\_swioama\_SLQSOMADMStartSession\_t::FwAvailability

8.999.2.2 swi\_uint256\_t unpack\_swioama\_SLQSOMADMStartSession\_t::ParamPresenceMask

## 8.1000 unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t Struct Reference

### Data Fields

- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1000.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1000.2 Field Documentation

8.1000.2.1 swi\_uint256\_t unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t::ParamPresenceMask

8.1000.2.2 uint16\_t unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t::Tlvresult

## 8.1001 unpack\_tmd\_SLQSTmdGetMitigationDevList\_t Struct Reference

### Data Fields

- uint8\_t [MitigationDevListLen](#)
- [tmd\\_mitigationDevList](#) [MitigationDevList](#) [255]
- uint16\_t [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1001.1 Detailed Description

This structure contains mitigation devices list from the remote endpoint

#### Parameters

<i>MitigationDevListLen</i>	<ul style="list-style-type: none"> <li>• Mitigation Device List Length (Optional)</li> <li>• Number of sets of the following elements <ul style="list-style-type: none"> <li>– MitigationDevList</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
-----------------------------	---

<i>pMitigationDev- List</i>	<ul style="list-style-type: none"> <li>• Mitigation Device List (Optional)</li> <li>• See <a href="#">tmd_mitigationDevList</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1001.2 Field Documentation

8.1001.2.1 `tmd_mitigationDevList unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevList[255]`

8.1001.2.2 `uint8_t unpack_tmd_SLQSTmdGetMitigationDevList_t::MitigationDevListLen`

8.1001.2.3 `swi_uint256_t unpack_tmd_SLQSTmdGetMitigationDevList_t::ParamPresenceMask`

8.1001.2.4 `uint16_t unpack_tmd_SLQSTmdGetMitigationDevList_t::Tlvresult`

## 8.1002 unpack\_tmd\_SLQSTmdGetMitigationLvl\_t Struct Reference

### Data Fields

- `uint8_t` [CurrentmitigationLvl](#)
- `uint8_t` [ReqMitigationLvl](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.1002.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>Currentmitigation- Lvl</i>	<ul style="list-style-type: none"> <li>• Current thermal mitigation level (Optional) <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>16</b></li> </ul> </li> </ul>
<i>ReqMitigationLvl</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. <ul style="list-style-type: none"> <li>– Bit to check in ParamPresenceMask - <b>17</b></li> </ul> </li> </ul>
<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1002.2 Field Documentation

8.1002.2.1 uint8\_t unpack\_tmd\_SLQSTmdGetMitigationLvl\_t::CurrentmitigationLvl

8.1002.2.2 swi\_uint256\_t unpack\_tmd\_SLQSTmdGetMitigationLvl\_t::ParamPresenceMask

8.1002.2.3 uint8\_t unpack\_tmd\_SLQSTmdGetMitigationLvl\_t::ReqMitigationLvl

8.1002.2.4 uint16\_t unpack\_tmd\_SLQSTmdGetMitigationLvl\_t::Tlvresult

## 8.1003 unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t Struct Reference

### Data Fields

- uint8\_t [deviceIDLen](#)
- char [deviceID](#) [255]
- uint8\_t [lvl](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1003.1 Detailed Description

Mitigation Level Report

#### Parameters

<i>deviceIDLen</i>	<ul style="list-style-type: none"> <li>• Mitigation Device ID Length</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>deviceID</i>	<ul style="list-style-type: none"> <li>• Mitigation Device ID</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>lvl</i>	<ul style="list-style-type: none"> <li>• Current thermal mitigation level</li> <li>• Bit to check in ParamPresenceMask - 2</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1003.2 Field Documentation

8.1003.2.1 char unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t::deviceID[255]

8.1003.2.2 uint8\_t unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t::deviceIDLen

8.1003.2.3 uint8\_t unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t::lvl

8.1003.2.4 swi\_uint256\_t unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t::ParamPresenceMask

## 8.1004 unpack\_tmd\_SLQSTmdRegNotMitigationLvl\_t Struct Reference

## Data Fields

- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1004.1 Detailed Description

This structure contains mitigation devices Level request parameters

#### Parameters

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• Unpack Result</li> <li>• Bit to check in ParamPresenceMask - <b>2</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1004.2 Field Documentation

8.1004.2.1 [swi\\_uint256\\_t unpack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t::ParamPresenceMask](#)

8.1004.2.2 [uint16\\_t unpack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t::Tlvresult](#)

## 8.1005 unpack\_uim\_ChangePin\_t Struct Reference

## Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint16\\_t Tlvresult](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1005.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.1005.2 Field Documentation

8.1005.2.1 `swi_uint256_t unpack_uim_ChangePin_t::ParamPresenceMask`

8.1005.2.2 `uim_encryptedPIN1* unpack_uim_ChangePin_t::pEncryptedPIN1`

8.1005.2.3 `uint32_t* unpack_uim_ChangePin_t::pIndicationToken`

8.1005.2.4 `uim_remainingRetries* unpack_uim_ChangePin_t::pRemainingRetries`

8.1005.2.5 `uint16_t unpack_uim_ChangePin_t::Tlvresult`

## 8.1006 unpack\_uim\_GetCardStatus\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)
- [uim\\_hotSwapStatus](#) \* [pHotSwapStatus](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1006.1 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_hotSwapStatus</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1006.2 Field Documentation

8.1006.2.1 `swi_uint256_t unpack_uim_GetCardStatus_t::ParamPresenceMask`

8.1006.2.2 `uim_cardStatus* unpack_uim_GetCardStatus_t::pCardStatus`

8.1006.2.3 `uim_hotSwapStatus* unpack_uim_GetCardStatus_t::pHotSwapStatus`

8.1006.2.4 `uint16_t unpack_uim_GetCardStatus_t::Tlvresult`

## 8.1007 unpack\_uim\_GetCardStatusV2\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* [pCardStatus](#)
- [uim\\_hotSwapStatus](#) \* [pHotSwapStatus](#)
- [uim\\_validCardStatus](#) \* [pValidCardStatus](#)
- [uim\\_simBusyStatus](#) \* [pSimBusyStatus](#)
- `uint16_t Tlvresult`
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1007.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHotSwap-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_hotSwapStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pValidCard-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_validCardStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pSimBusy-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_simBusyStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1007.2 Field Documentation

8.1007.2.1 `swi_uint256_t unpack_uim_GetCardStatusV2_t::ParamPresenceMask`

8.1007.2.2 `uim_cardStatus* unpack_uim_GetCardStatusV2_t::pCardStatus`

8.1007.2.3 `uim_hotSwapStatus* unpack_uim_GetCardStatusV2_t::pHotSwapStatus`

8.1007.2.4 `uim_simBusyStatus* unpack_uim_GetCardStatusV2_t::pSimBusyStatus`

8.1007.2.5 `uim_validCardStatus* unpack_uim_GetCardStatusV2_t::pValidCardStatus`

8.1007.2.6 `uint16_t unpack_uim_GetCardStatusV2_t::Tlvresult`

## 8.1008 unpack\_uim\_ReadTransparent\_t Struct Reference

### Data Fields

- [uim\\_cardResult](#) \* [pCardResult](#)
- [uim\\_readResult](#) \* [pReadResult](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint8\\_t](#) \* [pEncryptedData](#)
- [uint16\\_t](#) [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1008.1 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="#">uim_cardResult</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pReadResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_readResult</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1008.2 Field Documentation

8.1008.2.1 [swi\\_uint256\\_t](#) [unpack\\_uim\\_ReadTransparent\\_t::ParamPresenceMask](#)

8.1008.2.2 [uim\\_cardResult](#)\* [unpack\\_uim\\_ReadTransparent\\_t::pCardResult](#)

8.1008.2.3 [uint8\\_t](#)\* [unpack\\_uim\\_ReadTransparent\\_t::pEncryptedData](#)

8.1008.2.4 [uint32\\_t](#)\* [unpack\\_uim\\_ReadTransparent\\_t::pIndicationToken](#)

8.1008.2.5 [uim\\_readResult](#)\* [unpack\\_uim\\_ReadTransparent\\_t::pReadResult](#)

8.1008.2.6 [uint16\\_t](#) [unpack\\_uim\\_ReadTransparent\\_t::Tlvresult](#)

## 8.1009 unpack\_uim\_SetPinProtection\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [uint16\\_t](#) [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1009.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1009.2 Field Documentation

8.1009.2.1 [swi\\_uint256\\_t](#) [unpack\\_uim\\_SetPinProtection\\_t::ParamPresenceMask](#)

8.1009.2.2 [uim\\_encryptedPIN1](#)\* [unpack\\_uim\\_SetPinProtection\\_t::pEncryptedPIN1](#)

8.1009.2.3 [uint32\\_t](#)\* [unpack\\_uim\\_SetPinProtection\\_t::pIndicationToken](#)

8.1009.2.4 [uim\\_remainingRetries](#)\* [unpack\\_uim\\_SetPinProtection\\_t::pRemainingRetries](#)

8.1009.2.5 [uint16\\_t](#) [unpack\\_uim\\_SetPinProtection\\_t::Tlvresult](#)

## 8.1010 unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t Struct Reference

### Data Fields

- [slots\\_t](#) [slotsstatusChange](#)
- [uint8\\_t](#) [bNumberOfPhySlots](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)



### 8.1010.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>bNumberOfPhy-Slots</i>	<ul style="list-style-type: none"> <li>• Number of Physical Slot(s)</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1010.2 Field Documentation

8.1010.2.1 `uint8_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::bNumberOfPhySlots`

8.1010.2.2 `swi_uint256_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::ParamPresenceMask`

8.1010.2.3 `slots_t unpack_uim_SetUimSlotStatusChangeCallback_ind_t::slotsstatusChange`

## 8.1011 unpack\_uim\_SLQSUIAuthenticate\_t Struct Reference

### Data Fields

- `uim_cardResult * pCardResult`
- `uim_authenticateResult * pAuthenticateResult`
- `uint32_t * pIndicationToken`
- `swi_uint256_t ParamPresenceMask`

### 8.1011.1 Detailed Description

This structure contains information of the response parameters associated with a SLQSUIAuthenticate.

#### Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_cardResult</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pAuthenticate-Result(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_authenticateResult</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.1011.2 Field Documentation

8.1011.2.1 `swi_uint256_t unpack_uim_SLQSUIMAuthenticate_t::ParamPresenceMask`

8.1011.2.2 `uim_authenticateResult* unpack_uim_SLQSUIMAuthenticate_t::pAuthenticateResult`

8.1011.2.3 `uim_cardResult* unpack_uim_SLQSUIMAuthenticate_t::pCardResult`

8.1011.2.4 `uint32_t* unpack_uim_SLQSUIMAuthenticate_t::pIndicationToken`

## 8.1012 `unpack_uim_SLQSUIMDepersonalization_t` Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1012.1 Detailed Description

This structure contains information of the response parameters associated with a Depersonalization.

#### Parameters

<i>pRemainingRetries(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> <li>Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1012.2 Field Documentation

8.1012.2.1 `swi_uint256_t unpack_uim_SLQSUIMDepersonalization_t::ParamPresenceMask`

8.1012.2.2 `uim_remainingRetries* unpack_uim_SLQSUIMDepersonalization_t::pRemainingRetries`

## 8.1013 `unpack_uim_SLQSUIMEventRegister_t` Struct Reference

### Data Fields

- `uint32_t` [eventMask](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1013.1 Detailed Description

This structure contains unpack event register parameter.

## Parameters

<i>eventMask</i>	<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>
------------------	---

- Bit to check in ParamPresenceMask - 16

## Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.1013.2 Field Documentation

8.1013.2.1 uint32\_t unpack\_uim\_SLQSUIEventRegister\_t::eventMask

8.1013.2.2 swi\_uint256\_t unpack\_uim\_SLQSUIEventRegister\_t::ParamPresenceMask

## 8.1014 unpack\_uim\_SLQSUIMGetConfiguration\_t Struct Reference

## Data Fields

- uint8\_t \* [pAutoSelection](#)
- uim\_personalizationStatus \* [pPersonalizationStatus](#)
- uint8\_t \* [pHaltSubscription](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.1014.1 Detailed Description

This structure contains information of the response parameters associated with Get Configuration.

## Parameters

<i>pAuto-Selection(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> <li>• Bit to check in ParamPresenceMask - 16</li> </ul>
<i>p-Personalization-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_personalizationStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - 17</li> </ul>

<i>pHalt-Subscription(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> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1014.2 Field Documentation

8.1014.2.1 `swi_uint256_t unpack_uim_SLQSUIMGetConfiguration_t::ParamPresenceMask`

8.1014.2.2 `uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pAutoSelection`

8.1014.2.3 `uint8_t* unpack_uim_SLQSUIMGetConfiguration_t::pHaltSubscription`

8.1014.2.4 `uim_personalizationStatus* unpack_uim_SLQSUIMGetConfiguration_t::pPersonalizationStatus`

## 8.1015 `unpack_uim_SLQSUIMGetFileAttributes_t` Struct Reference

### Data Fields

- [uim\\_cardResult](#) \* [pCardResult](#)
- [uim\\_fileAttributes](#) \* [pFileAttributes](#)
- [uint32\\_t](#) \* [pIndicationToken](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1015.1 Detailed Description

This structure contains information of the response parameters associated with a Get File Attributes

#### Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_cardResult</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pFile-Attributes(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_fileAttributes</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></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> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1015.2 Field Documentation

8.1015.2.1 `swi_uint256_t` `unpack_uim_SLQSUIMGetFileAttributes_t::ParamPresenceMask`

8.1015.2.2 `uim_cardResult*` `unpack_uim_SLQSUIMGetFileAttributes_t::pCardResult`

8.1015.2.3 `uim_fileAttributes*` `unpack_uim_SLQSUIMGetFileAttributes_t::pFileAttributes`

8.1015.2.4 `uint32_t*` `unpack_uim_SLQSUIMGetFileAttributes_t::pIndicationToken`

## 8.1016 unpack\_uim\_SLQSUIMGetServiceStatus\_t Struct Reference

### Data Fields

- [uim\\_UIMGetFDNStatus](#) \* `pFDNStatus`
- [uim\\_UIMGetHiddenKeyStatus](#) \* `pHiddenKeyStatus`
- [uim\\_UIMGetIndex](#) \* `pIndex`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.1016.1 Detailed Description

This structure contains information of the response parameters associated with Get Services Status API.

#### Parameters

<i>pFDN-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMGetFDNStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pHiddenKey-Status(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMGetHiddenKeyStatus</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pIndex(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_UIMGetIndex</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not returned.

## 8.1016.2 Field Documentation

8.1016.2.1 `swi_uint256_t` `unpack_uim_SLQSUIMGetServiceStatus_t::ParamPresenceMask`

8.1016.2.2 `uim_UIMGetFDNStatus*` `unpack_uim_SLQSUIMGetServiceStatus_t::pFDNStatus`

8.1016.2.3 `uim_UIMGetHiddenKeyStatus*` `unpack_uim_SLQSUIMGetServiceStatus_t::pHiddenKeyStatus`

8.1016.2.4 `uim_UIMGetIndex*` `unpack_uim_SLQSUIMGetServiceStatus_t::pIndex`

## 8.1017 unpack\_uim\_SLQSUIMGetSlotsStatus\_t Struct Reference

### Data Fields

- [uint8\\_t](#) \* [pNumberOfPhySlot](#)
- [slots\\_t](#) \* [pUimSlotsStatus](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1017.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>pNumberOfPhySlot</i>	<ul style="list-style-type: none"> <li>• Number of sets of the Slot Status.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> <li>• Slots Status See <a href="#">slots_t</a> for more information..</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1017.2 Field Documentation

8.1017.2.1 [swi\\_uint256\\_t](#) [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t::ParamPresenceMask](#)

8.1017.2.2 [uint8\\_t](#)\* [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t::pNumberOfPhySlot](#)

8.1017.2.3 [slots\\_t](#)\* [unpack\\_uim\\_SLQSUIMGetSlotsStatus\\_t::pUimSlotsStatus](#)

## 8.1018 unpack\_uim\_SLQSUIMGetSlotsStatusV2\_t Struct Reference

### Data Fields

- [uim\\_GetSlotsStatusTlv](#) \* [pGetSlotsStatusTlv](#)
- [uim\\_GetSlotsInfoTlv](#) \* [pGetSlotsInfoTlv](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1018.1 Detailed Description

This structure contains information of the response parameters associated with a Get Slots Status API.

#### Parameters

<i>pNumberOfPhySlot</i>	<ul style="list-style-type: none"> <li>• Number of sets of the Slot Status.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
-------------------------	---

<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> <li>• Slots Status See <a href="#">slots_t</a> for more information..</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pNumberOfPhy-SlotInfo</i>	<ul style="list-style-type: none"> <li>• Number of Physical slot information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pUimSlotsStatus</i>	<ul style="list-style-type: none"> <li>• Slots Status See <a href="#">uim_physlotsInfo</a> for more information..</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1018.2 Field Documentation

8.1018.2.1 `swi_uint256_t unpack_uim_SLQSUIGetSlotsStatusV2_t::ParamPresenceMask`

8.1018.2.2 `uim_GetSlotsInfoTlv* unpack_uim_SLQSUIGetSlotsStatusV2_t::pGetSlotsInfoTlv`

8.1018.2.3 `uim_GetSlotsStatusTlv* unpack_uim_SLQSUIGetSlotsStatusV2_t::pGetSlotsStatusTlv`

## 8.1019 unpack\_uim\_SLQSUIReadRecord\_t Struct Reference

### Data Fields

- [uim\\_cardResultInfo](#) \* [pCardResult](#)
- [uim\\_readResultInfo](#) \* [pReadResult](#)
- [uim\\_additionalReadResult](#) \* [pAdditionalReadResult](#)
- [uim\\_indToken](#) \* [pIndicationToken](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1019.1 Detailed Description

This structure contains information of the response parameters associated with a Read Record API.

#### Parameters

<i>pCardResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_cardResultInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pReadResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_readResultInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAdditional-ReadResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_additionalReadResult</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_indToken</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
-----------------------------------	---

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.1019.2 Field Documentation

8.1019.2.1 `uim_additionalReadResult* unpack_uim_SLQSUIMReadRecord_t::pAdditionalReadResult`

8.1019.2.2 `swi_uint256_t unpack_uim_SLQSUIMReadRecord_t::ParamPresenceMask`

8.1019.2.3 `uim_cardResultInfo* unpack_uim_SLQSUIMReadRecord_t::pCardResult`

8.1019.2.4 `uim_indToken* unpack_uim_SLQSUIMReadRecord_t::pIndicationToken`

8.1019.2.5 `uim_readResultInfo* unpack_uim_SLQSUIMReadRecord_t::pReadResult`

### 8.1020 unpack\_uim\_SLQSUIMRefreshCallback\_Ind\_t Struct Reference

#### Data Fields

- `uint8_t TlvPresent`
- `uim_refreshevent refreshEvent`
- `swi_uint256_t ParamPresenceMask`

#### 8.1020.1 Detailed Description

This structure hold parameters about UIM refresh event indication.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if refresh event TLV is present in indication, otherwise 0</li> </ul>
<i>refreshEvent[OPTIONAL]</i>	<ul style="list-style-type: none"> <li>• see <a href="#">uim_refreshevent</a></li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1020.2 Field Documentation

8.1020.2.1 `swi_uint256_t unpack_uim_SLQSUIMRefreshCallback_Ind_t::ParamPresenceMask`

8.1020.2.2 `uim_refreshevent unpack_uim_SLQSUIMRefreshCallback_Ind_t::refreshEvent`



8.1020.2.3 uint8\_t unpack\_uim\_SLQSUIMRefreshCallback\_Ind\_t::TlvPresent

## 8.1021 unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t Struct Reference

### Data Fields

- [uim\\_refreshevent](#) \* pRefreshEvent
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1021.1 Detailed Description

This structure contains information of the response parameters associated with a Refresh Get Last Event.

#### Parameters

<i>refreshEvent(-Optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_refreshevent</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1021.2 Field Documentation

8.1021.2.1 [swi\\_uint256\\_t](#) unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t::ParamPresenceMask

8.1021.2.2 [uim\\_refreshevent](#)\* unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t::pRefreshEvent

## 8.1022 unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind\_t Struct Reference

### Data Fields

- [uim\\_cardStatus](#) \* pCardStatus
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1022.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1022.2 Field Documentation

8.1022.2.1 `swi_uint256_t unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::ParamPresenceMask`

8.1022.2.2 `uim_cardStatus* unpack_uim_SLQSUIMSetStatusChangeCallBack_ind_t::pCardStatus`

## 8.1023 `unpack_uim_SLQSUIMWriteRecord_t` Struct Reference

### Data Fields

- `uim_cardResultInfo * pCardResult`
- `uim_indToken * pIndicationToken`
- `swi_uint256_t ParamPresenceMask`

### 8.1023.1 Detailed Description

This structure contains information of the response parameters associated with a Write Record API.

#### Parameters

<i>pCardResult</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_cardResultInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pIndicationToken(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_indToken</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

#### Note

Using NULL for the pointers would make sure that the parameter is not added to the request.

### 8.1023.2 Field Documentation

8.1023.2.1 `swi_uint256_t unpack_uim_SLQSUIMWriteRecord_t::ParamPresenceMask`

8.1023.2.2 `uim_cardResultInfo* unpack_uim_SLQSUIMWriteRecord_t::pCardResult`

8.1023.2.3 `uim_indToken* unpack_uim_SLQSUIMWriteRecord_t::pIndicationToken`

## 8.1024 `unpack_uim_SLQSUIMWriteTransparent_t` Struct Reference

### Data Fields

- `uim_cardResultInfo * pCardResult`
- `uim_indToken * pIndicationToken`
- `swi_uint256_t ParamPresenceMask`

### 8.1024.1 Detailed Description

This structure contains information of the response parameters associated with a Write Transparent API.

## Parameters

<i>pCard-Result(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_cardResultInfo</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_indToken</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## Note

Using NULL for the pointers would make sure that the parameter is not returned.

## 8.1024.2 Field Documentation

8.1024.2.1 `swi_uint256_t unpack_uim_SLQSUIMWriteTransparent_t::ParamPresenceMask`

8.1024.2.2 `uim_cardResultInfo* unpack_uim_SLQSUIMWriteTransparent_t::pCardResult`

8.1024.2.3 `uim_indToken* unpack_uim_SLQSUIMWriteTransparent_t::pIndicationToken`

## 8.1025 unpack\_uim\_UnblockPin\_t Struct Reference

## Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- `uint32_t` \* [pIndicationToken](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.1025.1 Detailed Description

This structure contains information of the response parameters associated with a set of PIN related API's.

## Parameters

<i>pRemaining-Retries(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_remainingRetries</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pIndication-Token(optional)</i>	<ul style="list-style-type: none"> <li>Response in Indication.</li> <li>When this TLV is present, it indicates that the result is provided in a subsequent indication.</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

### 8.1025.2 Field Documentation

8.1025.2.1 `swi_uint256_t unpack_uim_UnblockPin_t::ParamPresenceMask`

8.1025.2.2 `uim_encryptedPIN1* unpack_uim_UnblockPin_t::pEncryptedPIN1`

8.1025.2.3 `uint32_t* unpack_uim_UnblockPin_t::pIndicationToken`

8.1025.2.4 `uim_remainingRetries* unpack_uim_UnblockPin_t::pRemainingRetries`

8.1025.2.5 `uint16_t unpack_uim_UnblockPin_t::Tlvresult`

## 8.1026 unpack\_uim\_UnblockPinV2\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- `uint32_t` \* [pIndicationToken](#)
- [uim\\_cardResult](#) \* [pCardResult](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1026.1 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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_encryptedPIN1</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></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>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pCardResult(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">uim_cardResult</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1026.2 Field Documentation

- 8.1026.2.1 `swi_uint256_t unpack_uim_UnblockPinV2_t::ParamPresenceMask`
- 8.1026.2.2 `uim_cardResult* unpack_uim_UnblockPinV2_t::pCardResult`
- 8.1026.2.3 `uim_encryptedPIN1* unpack_uim_UnblockPinV2_t::pEncryptedPIN1`
- 8.1026.2.4 `uint32_t* unpack_uim_UnblockPinV2_t::pIndicationToken`
- 8.1026.2.5 `uim_remainingRetries* unpack_uim_UnblockPinV2_t::pRemainingRetries`
- 8.1026.2.6 `uint16_t unpack_uim_UnblockPinV2_t::Tlvresult`

## 8.1027 unpack\_uim\_VerifyPin\_t Struct Reference

### Data Fields

- [uim\\_remainingRetries](#) \* [pRemainingRetries](#)
- [uim\\_encryptedPIN1](#) \* [pEncryptedPIN1](#)
- `uint32_t` \* [pIndicationToken](#)
- `uint16_t` [Tlvresult](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1027.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pEncryptedPIN1(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">uim_encryptedPIN1</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1027.2 Field Documentation

- 8.1027.2.1 `swi_uint256_t unpack_uim_VerifyPin_t::ParamPresenceMask`
- 8.1027.2.2 `uim_encryptedPIN1* unpack_uim_VerifyPin_t::pEncryptedPIN1`

8.1027.2.3 `uint32_t* unpack_uim_VerifyPin_t::pIndicationToken`

8.1027.2.4 `uim_remainingRetries* unpack_uim_VerifyPin_t::pRemainingRetries`

8.1027.2.5 `uint16_t unpack_uim_VerifyPin_t::Tlvresult`

## 8.1028 `unpack_voice_allCallStatusCallback_ind_t` Struct Reference

### Data Fields

- [voice\\_arrCallInfo](#) `arrCallInfomation`
- [voice\\_arrRemotePartyNum](#) \* `pArrRemotePartyNum`
- [voice\\_arrRemotePartyName](#) \* `pArrRemotePartyName`
- [voice\\_arrAlertingType](#) \* `pArrAlertingType`
- [voice\\_arrSvcOption](#) \* `pArrSvcOption`
- [voice\\_arrCallEndReason](#) \* `pArrCallEndReason`
- [voice\\_arrAlphaID](#) \* `pArrAlphaID`
- [voice\\_arrConnectPartyNum](#) \* `pArrConnectPartyNum`
- [voice\\_arrDiagInfo](#) \* `pArrDiagInfo`
- [voice\\_arrCalledPartyNum](#) \* `pArrCalledPartyNum`
- [voice\\_arrRedirPartyNum](#) \* `pArrRedirPartyNum`
- [voice\\_arrAlertingPattern](#) \* `pArrAlertingPattern`
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.1028.1 Detailed Description

This structure contains information about the indication parameters for all the calls originating or terminating from a particular device.

#### Parameters

<i>arrCall-Info(mandatory)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrCallInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pArrRemote-Party-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrRemotePartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pArrRemote-Party-Name(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrRemotePartyName</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pArrAlerting-Type(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrAlertingType</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pArrSvc-Option(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrSvcOption</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pArrCallEnd-Reason(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrCallEndReason</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>

<i>pArrAlphaID(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrAlphaID</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pArrConnectPartyNum(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrConnectPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pArrDiagInfo(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrDiagInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pArrCalledPartyNum(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrCalledPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pArrRedirPartyNum(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrRedirPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pArrAlertingPattern(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrAlertingPattern</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1028.2 Field Documentation

8.1028.2.1 `voice_arrCallInfo` `unpack_voice_allCallStatusCallback_ind_t::arrCallInformation`

8.1028.2.2 `swi_uint256_t` `unpack_voice_allCallStatusCallback_ind_t::ParamPresenceMask`

8.1028.2.3 `voice_arrAlertingPattern*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlertingPattern`

8.1028.2.4 `voice_arrAlertingType*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlertingType`

8.1028.2.5 `voice_arrAlphaID*` `unpack_voice_allCallStatusCallback_ind_t::pArrAlphaID`

8.1028.2.6 `voice_arrCalledPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrCalledPartyNum`

8.1028.2.7 `voice_arrCallEndReason*` `unpack_voice_allCallStatusCallback_ind_t::pArrCallEndReason`

8.1028.2.8 `voice_arrConnectPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrConnectPartyNum`

8.1028.2.9 `voice_arrDiagInfo*` `unpack_voice_allCallStatusCallback_ind_t::pArrDiagInfo`

8.1028.2.10 `voice_arrRedirPartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrRedirPartyNum`

8.1028.2.11 `voice_arrRemotePartyName*` `unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyName`

8.1028.2.12 `voice_arrRemotePartyNum*` `unpack_voice_allCallStatusCallback_ind_t::pArrRemotePartyNum`

8.1028.2.13 `voice_arrSvcOption* unpack_voice_allCallStatusCallback_ind_t::pArrSvcOption`

## 8.1029 `unpack_voice_DTMFEventCallback_ind_t` Struct Reference

### Data Fields

- [voice\\_DTMFInfo](#) DTMFInformation
- `uint8_t * pOnLength`
- `uint8_t * pOffLength`
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1029.1 Detailed Description

This structure contains the parameters passed for DTMF event indication by the device.

#### Parameters

<i>DTMF- Information(mandatory)</i>	See <a href="#">voice_DTMFInfo</a> for more information. <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence- Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

#### Note

None

### 8.1029.2 Field Documentation

8.1029.2.1 `voice_DTMFInfo` `unpack_voice_DTMFEventCallback_ind_t::DTMFInformation`



8.1029.2.2 `swi_uint256_t` unpack\_voice\_DTMFEventCallback\_ind\_t::ParamPresenceMask

8.1029.2.3 `uint8_t*` unpack\_voice\_DTMFEventCallback\_ind\_t::pOffLength

8.1029.2.4 `uint8_t*` unpack\_voice\_DTMFEventCallback\_ind\_t::pOnLength

## 8.1030 unpack\_voice\_OTASPStatusCallback\_ind\_t Struct Reference

### Data Fields

- `uint8_t` callID
- `uint8_t` OTASPStatus
- `swi_uint256_t` ParamPresenceMask

### 8.1030.1 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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>OTASPStatus</i>	<ul style="list-style-type: none"> <li>• OTASP status for the OTASP call. Values: <ul style="list-style-type: none"> <li>– 0x00 - OTASP_STATUS_SPL_UNLOCKED. SPL unlocked; only for user-initiated OTASP</li> <li>– 0x01 - OTASP_STATUS_SPRC_RETRIES_EXCEEDED. SPC retries exceeded; only for user-initiated OTASP</li> <li>– 0x02 - OTASP_STATUS_AKEY_EXCHANGED. A-key exchanged; only for user-initiated OTASP</li> <li>– 0x03 - OTASP_STATUS_SSD_UPDATED. SSD updated; for both user-initiated OTASP and network-initiated OTASP (OTAPA)</li> <li>– 0x04 - OTASP_STATUS_NAM_DOWNLOADED - NAM downloaded; only for user-initiated OTASP</li> <li>– 0x05 - OTASP_STATUS_MDN_DOWNLOADED - MDN downloaded; only for user-initiated OTASP</li> <li>– 0x06 - OTASP_STATUS_IMSI_DOWNLOADED - IMSI downloaded; only for user-initiated OTASP</li> <li>– 0x07 - OTASP_STATUS_PRL_DOWNLOADED - PRL downloaded; only for user-initiated OTASP</li> <li>– 0x08 - OTASP_STATUS_COMMITTED - Commit successful; only for user-initiated OTASP</li> <li>– 0x09 - OTASP_STATUS_OTAPA_STARTED - OTAPA started; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0A - OTASP_STATUS_OTAPA_STOPPED - OTAPA stopped; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0B - OTASP_STATUS_OTAPA_ABORTED - OTAPA aborted; only for network-initiated OTASP(OTAPA)</li> <li>– 0x0C - OTASP_STATUS_OTAPA_COMMITTED - OTAPA committed; only for network-initiated OTASP(OTAPA)</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
--------------------------	--

## 8.1030.2 Field Documentation

8.1030.2.1 `uint8_t unpack_voice_OTASPStatusCallback_ind_t::callID`

8.1030.2.2 `uint8_t unpack_voice_OTASPStatusCallback_ind_t::OTASPStatus`

8.1030.2.3 `swi_uint256_t unpack_voice_OTASPStatusCallback_ind_t::ParamPresenceMask`

## 8.1031 `unpack_voice_SLQSOriinateUSSD_t` Struct Reference

### Data Fields

- `uint16_t * pfailureCause`
- `voice_alphaIDInfo * pAlphaIDInfo`
- `struct voice_USSInfo * pUSSDInfo`
- `uint8_t * pCcResultType`
- `uint8_t * pCallId`
- `voice_ccSUPSType * pCCSuppsType`
- `swi_uint256_t ParamPresenceMask`

### 8.1031.1 Detailed Description

This structure contains the parameters of USS response.

---

## Parameters

---

<i>pfailureCause</i>	<ul style="list-style-type: none"> <li>• Failure Cause.</li> <li>• 0 - QMI_FAILURE_CAUSE_OFFLINE - Phone is offline</li> <li>• 20 - QMI_FAILURE_CAUSE_CDMA_LOCK - Phone is CDMA locked until a power cycle; CDMA only</li> <li>• 21 - QMI_FAILURE_CAUSE_NO_SRV - Phone has no service</li> <li>• 22 - QMI_FAILURE_CAUSE_FADE - Call has ended abnormally</li> <li>• 23 - QMI_FAILURE_CAUSE_INTERCEPT - Received intercept from the base station; originating only; CDMA only</li> <li>• 24 - QMI_FAILURE_CAUSE_REORDER - Received reorder from the base station; originating only; CDMA only</li> <li>• 25 - QMI_FAILURE_CAUSE_REL_NORMAL - Received release from the base station; no reason was given</li> <li>• 26 - QMI_FAILURE_CAUSE_REL_SO_REJ - Received release from the base station; SO reject; CDMA only</li> <li>• 27 - QMI_FAILURE_CAUSE_INCOM_CALL - Received incoming call from the base station</li> <li>• 28 - QMI_FAILURE_CAUSE_ALERT_STOP - Received alert stop from the base station; incoming only; CDMA only</li> <li>• 29 - QMI_FAILURE_CAUSE_CLIENT_END - Client ended the call</li> <li>• 30 - QMI_FAILURE_CAUSE_ACTIVATION - Received end activation; OTASP call only; CDMA only</li> <li>• 31 - QMI_FAILURE_CAUSE_MC_ABORT - MC aborted the origination/conversation; CDMA only</li> <li>• 32 - QMI_FAILURE_CAUSE_MAX_ACCESS_PROBE - Maximum access probes were transmitted; CDMA only</li> <li>• 33 - QMI_FAILURE_CAUSE_PSIST_N - Persistence test failure; FEATURE_JCDMA only; CDMA only</li> <li>• 34 - QMI_FAILURE_CAUSE_UIM_NOT_PRESENT - R-UIM is not present</li> <li>• 35 - QMI_FAILURE_CAUSE_ACC_IN_PROG - Access attempt is already in progress</li> <li>• 36 - QMI_FAILURE_CAUSE_ACC_FAIL - Access failure for a reason other than the above</li> <li>• 37 - QMI_FAILURE_CAUSE_RETRY_ORDER - Received retry order; originating only; IS 2000; CDMA only</li> <li>• 38 - QMI_FAILURE_CAUSE_CCS_NOT_SUPPORTED_BYBS - Concurrent service is not supported by the base station</li> <li>• 39 - QMI_FAILURE_CAUSE_NO_RESPONSE_FROM_BS - No response was received from the base station</li> <li>• 40 - QMI_FAILURE_CAUSE_REJECTED_BY_BS - Call was rejected by the base station; CDMA only</li> <li>• 41 - QMI_FAILURE_CAUSE_INCOMPATIBLE - Concurrent services requested were not compatible; CDMA only</li> <li>• 42 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station; CDMA only</li> <li>• 43 - QMI_FAILURE_CAUSE_ALREADY_IN_TC - Corresponds to CM_CALL_ORIGERR_ALREADY_IN_TC</li> <li>• 44 - QMI_FAILURE_CAUSE_EMERGENCY_FLASHED - Call is ended because an emergency call was flashed over this call; CDMA only</li> <li>• 45 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_GPS - Used if CM is ending a GPS call in preference of a user call</li> <li>• 46 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_SMS - Used if CM is ending an SMS call in preference of a user call</li> <li>• 47 - QMI_FAILURE_CAUSE_USER_CALL_ORIGDURING_DATA - Used if CM is ending a data call in preference of an emergency call</li> <li>• 48 - QMI_FAILURE_CAUSE_REDIR_OR_HANDOFF - Call was rejected because of a redirection or handoff</li> </ul>
	<ul style="list-style-type: none"> <li>• 49 - QMI_FAILURE_CAUSE_ACCESS_BLOCK - Access is blocked by the base station for all mobiles; KDDI-specific; CDMA only</li> <li>• 50 - QMI_FAILURE_CAUSE_OTASP_SPC_ERR - To support OTASP SPC Error indication</li> <li>• 51 - QMI_FAILURE_CAUSE_IS707B_MAX_ACC - Maximum access probes for an IS-707B</li> </ul>

<i>AlphaIdentifier</i>	<ul style="list-style-type: none"> <li>• see <a href="#">voice_alphaIDInfo</a> definition</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pUSSDInfo</i>	<ul style="list-style-type: none"> <li>• USS Data from Network (See structure <a href="#">voice_USSInfo</a>)</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pCcResultType</i>	<ul style="list-style-type: none"> <li>• CC result code</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pCallId</i>	<ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> <li>• NULL pointer - Invalid data.</li> </ul>
<i>pCCSuppsType</i>	<ul style="list-style-type: none"> <li>• See structure '<a href="#">voice_ccSUPSType</a>' definition</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1031.2 Field Documentation

8.1031.2.1 [voice\\_alphaIDInfo\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pAlphaIDInfo](#)

8.1031.2.2 [swi\\_uint256\\_t](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::ParamPresenceMask](#)

8.1031.2.3 [uint8\\_t\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pCallId](#)

8.1031.2.4 [uint8\\_t\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pCcResultType](#)

8.1031.2.5 [voice\\_ccSUPSType\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pCCSuppsType](#)

8.1031.2.6 [uint16\\_t\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pfailureCause](#)

8.1031.2.7 [struct voice\\_USSInfo\\*](#) [unpack\\_voice\\_SLQSOriinateUSSD\\_t::pUSSDInfo](#)

## 8.1032 unpack\_voice\_SLQSVoiceAnswerCall\_t Struct Reference

### Data Fields

- [uint8\\_t \\*](#) [pCallId](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1032.1 Detailed Description

Contains the parameters passed for pack voice Answer Call.

## Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> <li>Unique call identifier for the call that must be answered.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1032.2 Field Documentation

8.1032.2.1 `swi_uint256_t unpack_voice_SLQSVoiceAnswerCall_t::ParamPresenceMask`8.1032.2.2 `uint8_t* unpack_voice_SLQSVoiceAnswerCall_t::pCallId`8.1033 `unpack_voice_SLQSVoiceBurstDTMF_t` Struct Reference

## Data Fields

- `uint8_t * pCallID`
- `swi_uint256_t ParamPresenceMask`

## 8.1033.1 Detailed Description

This structure contains Voice Burst DTMF Information for unpack.

## Parameters

<i>pCallID</i>	<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> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1033.2 Field Documentation

8.1033.2.1 `swi_uint256_t unpack_voice_SLQSVoiceBurstDTMF_t::ParamPresenceMask`8.1033.2.2 `uint8_t* unpack_voice_SLQSVoiceBurstDTMF_t::pCallID`8.1034 `unpack_voice_SLQSVoiceDialCall_t` Struct Reference

## Data Fields

- `uint8_t * pCallID`
- `voice_alphaIDInfo * pAlphaIDInfo`

- [uint8\\_t \\* pCCResultType](#)
- [voice\\_ccSUPSType \\* pCCSUPSType](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1034.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pAlphaID-Info(optional)</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pCCSUPS-Type(optional)</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of ccSUPSType</li> <li>• Data is present when pCCResultType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1034.2 Field Documentation

8.1034.2.1 [voice\\_alphaIDInfo\\*](#) [unpack\\_voice\\_SLQSVoiceDialCall\\_t::pAlphaIDInfo](#)

8.1034.2.2 [swi\\_uint256\\_t](#) [unpack\\_voice\\_SLQSVoiceDialCall\\_t::ParamPresenceMask](#)

8.1034.2.3 [uint8\\_t\\*](#) [unpack\\_voice\\_SLQSVoiceDialCall\\_t::pCallID](#)

8.1034.2.4 [uint8\\_t\\*](#) [unpack\\_voice\\_SLQSVoiceDialCall\\_t::pCCResultType](#)

8.1034.2.5 [voice\\_ccSUPSType\\*](#) [unpack\\_voice\\_SLQSVoiceDialCall\\_t::pCCSUPSType](#)

## 8.1035 unpack\_voice\_SLQSVoiceEndCall\_t Struct Reference

### Data Fields

- [uint8\\_t \\* pCallId](#)

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1035.1 Detailed Description

This structure contains unpack voice end call parameter.

#### Parameters

<i>pCallId</i>	<ul style="list-style-type: none"> <li>• Unique call identifier for the call that must be ended</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1035.2 Field Documentation

8.1035.2.1 [swi\\_uint256\\_t unpack\\_voice\\_SLQSVoiceEndCall\\_t::ParamPresenceMask](#)

8.1035.2.2 [uint8\\_t\\* unpack\\_voice\\_SLQSVoiceEndCall\\_t::pCallId](#)

## 8.1036 unpack\_voice\_SLQSVoiceGetAllCallInfo\_t Struct Reference

#### Data Fields

- [voice\\_arrCallInfo \\* pArrCallInfo](#)
- [voice\\_arrRemotePartyNum \\* pArrRemotePartyNum](#)
- [voice\\_arrRemotePartyName \\* pArrRemotePartyName](#)
- [voice\\_arrAlertingType \\* pArrAlertingType](#)
- [voice\\_arrUUSInfo \\* pArrUUSInfo](#)
- [voice\\_arrSvcOption \\* pArrSvcOption](#)
- [uint8\\_t \\* pOTASPStatus](#)
- [uint8\\_t \\* pVoicePrivacy](#)
- [voice\\_arrCallEndReason \\* pArrCallEndReason](#)
- [voice\\_arrAlphaID \\* pArrAlphaID](#)
- [voice\\_arrConnectPartyNum \\* pArrConnectPartyNum](#)
- [voice\\_arrDiagInfo \\* pArrDiagInfo](#)
- [voice\\_arrCalledPartyNum \\* pArrCalledPartyNum](#)
- [voice\\_arrRedirPartyNum \\* pArrRedirPartyNum](#)
- [voice\\_arrAlertingPattern \\* pArrAlertingPattern](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1036.1 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="#">voice_arrCallInfo</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>16</b></li></ul>
<i>pArrRemote-Party-Num(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_arrRemotePartyNum</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>17</b></li></ul>
<i>pArrRemote-Party-Name(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_arrRemotePartyName</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>18</b></li></ul>
<i>pArrAlerting-Type(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_arrAlertingType</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>19</b></li></ul>
<i>pArrUUS-Info(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_arrUUSInfo</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>20</b></li></ul>
<i>pArrSvc-Option(optional)</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_arrSvcOption</a> for more information.</li><li>• Bit to check in ParamPresenceMask - <b>21</b></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_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> <li>• Bit to check in ParamPresenceMask - <b>22</b></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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pArrCallEndReason(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrCallEndReason</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pArrAlphaID(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrAlphaID</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pArrConnectPartyNum(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrConnectPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>

<i>pArrDiag-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrDiagInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pArrCalledParty-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrCalledPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pArrRedirParty-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrRedirPartyNum</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pArrAlerting-Pattern(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_arrAlertingPattern</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1036.2 Field Documentation

8.1036.2.1 `swi_uint256_t unpack_voice_SLQSVoiceGetAllCallInfo_t::ParamPresenceMask`

8.1036.2.2 `voice_arrAlertingPattern* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingPattern`

8.1036.2.3 `voice_arrAlertingType* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlertingType`

8.1036.2.4 `voice_arrAlphaID* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrAlphaID`

8.1036.2.5 `voice_arrCalledPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCalledPartyNum`

8.1036.2.6 `voice_arrCallEndReason* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallEndReason`

8.1036.2.7 `voice_arrCallInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrCallInfo`

8.1036.2.8 `voice_arrConnectPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrConnectPartyNum`

8.1036.2.9 `voice_arrDiagInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrDiagInfo`

8.1036.2.10 `voice_arrRedirPartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRedirPartyNum`

8.1036.2.11 `voice_arrRemotePartyName* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyName`

8.1036.2.12 `voice_arrRemotePartyNum* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrRemotePartyNum`

8.1036.2.13 `voice_arrSvcOption* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrSvcOption`

8.1036.2.14 `voice_arrUUSInfo* unpack_voice_SLQSVoiceGetAllCallInfo_t::pArrUUSInfo`

8.1036.2.15 `uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pOTASPStatus`

8.1036.2.16 `uint8_t* unpack_voice_SLQSVoiceGetAllCallInfo_t::pVoicePrivacy`

## 8.1037 unpack\_voice\_SLQSVoiceGetCallBarring\_t Struct Reference

### Data Fields

- uint8\_t \* [pSvcClass](#)
- uint16\_t \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- uint8\_t \* [pCCResType](#)
- uint8\_t \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1037.1 Detailed Description

This structure contains Voice Get Call Barring Response Parameters

#### Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1037.2 Field Documentation

8.1037.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCallBarring_t::pAlphaIDInfo`

8.1037.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCallBarring_t::ParamPresenceMask`

8.1037.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCallID`

8.1037.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCCResType`

8.1037.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCallBarring_t::pCCSUPSType`

8.1037.2.6 `uint16_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pFailCause`

8.1037.2.7 `uint8_t*` `unpack_voice_SLQSVoiceGetCallBarring_t::pSvcClass`

## 8.1038 unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t Struct Reference

### Data Fields

- [voice\\_getCallFWInfo](#) \* `pGetCallFWInfo`
- `uint16_t` \* `pFailCause`
- [voice\\_alphaIDInfo](#) \* `pAlphaIDInfo`
- `uint8_t` \* `pCCResType`
- `uint8_t` \* `pCallID`
- [voice\\_ccSUPSType](#) \* `pCCSUPSType`
- [voice\\_getCallFWExtInfo](#) \* `pGetCallFWExtInfo`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1038.1 Detailed Description

This structure contains Voice Get Call Forwarding Status Unpack Parameters

#### Parameters

<i>pGetCallFWInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of getCallFWInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_getCallFWInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pGetCallFWExt-Info</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of getCallFWExtInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_getCallFWExtInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1038.2 Field Documentation

8.1038.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pAlphaIDInfo`

8.1038.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::ParamPresenceMask`

8.1038.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCallID`

8.1038.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCResType`

8.1038.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCallForwardingStatus_t::pCCSUPSType`

8.1038.2.6 uint16\_t\* unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t::pFailCause

8.1038.2.7 voice\_getCallFWExtInfo\* unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t::pGetCallFWExtInfo

8.1038.2.8 voice\_getCallFWInfo\* unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t::pGetCallFWInfo

## 8.1039 unpack\_voice\_SLQSVoiceGetCallInfo\_t Struct Reference

### Data Fields

- [voice\\_callInfo](#) \* [pCallInfo](#)
- [voice\\_remotePartyNum](#) \* [pRemotePartyNum](#)
- uint16\_t \* [pSrvOpt](#)
- uint8\_t \* [pVoicePrivacy](#)
- uint8\_t \* [pOTASPStatus](#)
- [voice\\_remotePartyName](#) \* [pRemotePartyName](#)
- [voice\\_UUSInfo](#) \* [pUUSInfo](#)
- uint8\_t \* [pAlertType](#)
- [voice\\_alphalDInfo](#) \* [pAlphalDInfo](#)
- [voice\\_connectNumInfo](#) \* [pConnectNumInfo](#)
- [voice\\_diagInfo](#) \* [pDiagInfo](#)
- uint32\_t \* [pAlertingPattern](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1039.1 Detailed Description

This structure contains information of the unpack parameters associated with a call.

#### Parameters

<i>pCall-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_callInfo</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pRemoteParty-Num(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_remotePartyNum</a> for more information.</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></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_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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>pRemoteParty-Name(optional)</i>	<ul style="list-style-type: none"> <li>• Applicable only for 3GPP devices.</li> <li>• See <a href="#">voice_remotePartyName</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pUUS-Info(optional)</i>	<ul style="list-style-type: none"> <li>• Applicable only for 3GPP devices.</li> <li>• See <a href="#">voice_UUSInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>



<i>pAlphaID-Info(optional)</i>	<ul style="list-style-type: none"> <li>• Applicable only for 3GPP devices.</li> <li>• See <a href="#">voice_alphaIDInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pConnectNum-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_connectNumInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pDiag-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_diagInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1039.2 Field Documentation

8.1039.2.1 `uint32_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertingPattern`

8.1039.2.2 `uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pAlertType`

8.1039.2.3 `voice_alphaIDInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pAlphaIDInfo`

8.1039.2.4 `swi_uint256_t unpack_voice_SLQSVoiceGetCallInfo_t::ParamPresenceMask`

8.1039.2.5 `voice_callInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pCallInfo`

8.1039.2.6 `voice_connectNumInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pConnectNumInfo`

8.1039.2.7 `voice_diagInfo* unpack_voice_SLQSVoiceGetCallInfo_t::pDiagInfo`

8.1039.2.8 `uint8_t* unpack_voice_SLQSVoiceGetCallInfo_t::pOTASPStatus`

8.1039.2.9 `voice_remotePartyName* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyName`

8.1039.2.10 `voice_remotePartyNum* unpack_voice_SLQSVoiceGetCallInfo_t::pRemotePartyNum`

8.1039.2.11 uint16\_t\* unpack\_voice\_SLQSVoiceGetCallInfo\_t::pSrvOpt

8.1039.2.12 voice\_UUSInfo\* unpack\_voice\_SLQSVoiceGetCallInfo\_t::pUUSInfo

8.1039.2.13 uint8\_t\* unpack\_voice\_SLQSVoiceGetCallInfo\_t::pVoicePrivacy

## 8.1040 unpack\_voice\_SLQSVoiceGetCallWaiting\_t Struct Reference

### Data Fields

- uint8\_t \* [pSvcClass](#)
- uint16\_t \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- uint8\_t \* [pCCResType](#)
- uint8\_t \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1040.1 Detailed Description

This structure contains Voice Get Call Waiting Unpack Parameters

#### Parameters

<i>pSvcClass</i>	<ul style="list-style-type: none"> <li>• Service class is a combination (sum) of information class constants (optional)</li> <li>• See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>• Service Class is set to 0 if call waiting is not active for any of the information classes.</li> <li>• 0xFF,if Not Available</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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 pCCRResType is present and is Voice.</li> <li>• If zero(0) then invalid.</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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 pCCRResType is present and is other than Voice. <ul style="list-style-type: none"> <li>– See <a href="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1040.2 Field Documentation

8.1040.2.1 [voice\\_alphaIDInfo](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pAlphaIDInfo](#)

8.1040.2.2 [swi\\_uint256\\_t](#) [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::ParamPresenceMask](#)

8.1040.2.3 [uint8\\_t](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pCallID](#)

8.1040.2.4 [uint8\\_t](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pCCRResType](#)

8.1040.2.5 [voice\\_ccSUPSType](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pCCSUPSType](#)

8.1040.2.6 [uint16\\_t](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pFailCause](#)

8.1040.2.7 [uint8\\_t](#)\* [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t::pSvcClass](#)

## 8.1041 unpack\_voice\_SLQSVoiceGetCLIP\_t Struct Reference

### Data Fields

- [voice\\_CLIPResp](#) \* [pCLIPResp](#)
- [uint16\\_t](#) \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [uint8\\_t](#) \* [pCCRResType](#)
- [uint8\\_t](#) \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1041.1 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 CLIPResp (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_CLIPResp</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1041.2 Field Documentation

8.1041.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCLIP_t::pAlphaIDInfo`8.1041.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCLIP_t::ParamPresenceMask`8.1041.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCLIP_t::pCallID`8.1041.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCLIP_t::pCCResType`8.1041.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCLIP_t::pCCSUPSType`

8.1041.2.6 `voice_CLIPResp`\* `unpack_voice_SLQSVoiceGetCLIP_t::pCLIPResp`

8.1041.2.7 `uint16_t`\* `unpack_voice_SLQSVoiceGetCLIP_t::pFailCause`

## 8.1042 unpack\_voice\_SLQSVoiceGetCLIR\_t Struct Reference

### Data Fields

- `voice_CLIRResp` \* `pCLIRResp`
- `uint16_t` \* `pFailCause`
- `voice_alphalDInfo` \* `pAlphaDInfo`
- `uint8_t` \* `pCCResType`
- `uint8_t` \* `pCallID`
- `voice_ccSUPSType` \* `pCCSUPSType`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1042.1 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 CLIRResp (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_CLIPResp</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphalDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1042.2 Field Documentation

8.1042.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCLIR_t::pAlphaIDInfo`

8.1042.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCLIR_t::ParamPresenceMask`

8.1042.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCLIR_t::pCallID`

8.1042.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCLIR_t::pCCResType`

8.1042.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCLIR_t::pCCSUPSType`

8.1042.2.6 `voice_CLIRResp*` `unpack_voice_SLQSVoiceGetCLIR_t::pCLIRResp`

8.1042.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCLIR_t::pFailCause`

## 8.1043 `unpack_voice_SLQSVoiceGetCNAP_t` Struct Reference

### Data Fields

- [voice\\_CNAPResp](#) \* `pCNAPResp`
- `uint16_t` \* `pFailCause`
- [voice\\_alphaIDInfo](#) \* `pAlphaIDInfo`
- `uint8_t` \* `pCCResType`
- `uint8_t` \* `pCallID`
- [voice\\_ccSUPSType](#) \* `pCCSUPSType`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1043.1 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 CNAPResp (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_CNAPResp</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1043.2 Field Documentation

8.1043.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCNAP_t::pAlphaIDInfo`

8.1043.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCNAP_t::ParamPresenceMask`

8.1043.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pCallID`

8.1043.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pCCResType`

8.1043.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCNAP_t::pCCSUPSType`

8.1043.2.6 `voice_CNAPResp*` `unpack_voice_SLQSVoiceGetCNAP_t::pCNAPResp`

8.1043.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCNAP_t::pFailCause`

## 8.1044 unpack\_voice\_SLQSVoiceGetCOLP\_t Struct Reference

### Data Fields

- [voice\\_COLPResp](#) \* [pCOLPResp](#)
- [uint16\\_t](#) \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- [uint8\\_t](#) \* [pCCResType](#)
- [uint8\\_t](#) \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1044.1 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 COLPResp (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_COLPResp</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>



<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.1044.2 Field Documentation

8.1044.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCOLP_t::pAlphaIDInfo`

8.1044.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCOLP_t::ParamPresenceMask`

8.1044.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pCallID`

8.1044.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pCCResType`

8.1044.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCOLP_t::pCCSUPSType`

8.1044.2.6 `voice_COLPResp*` `unpack_voice_SLQSVoiceGetCOLP_t::pCOLPResp`

8.1044.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCOLP_t::pFailCause`

## 8.1045 unpack\_voice\_SLQSVoiceGetCOLR\_t Struct Reference

### Data Fields

- `voice_COLRResp` \* `pCOLRResp`
- `uint16_t` \* `pFailCause`
- `voice_alphaIDInfo` \* `pAlphaIDInfo`
- `uint8_t` \* `pCCResType`
- `uint8_t` \* `pCallID`
- `voice_ccSUPSType` \* `pCCSUPSType`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1045.1 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 COLRResp (optional) <ul style="list-style-type: none"> <li>See <a href="#">voice_COLRResp</a> for more information</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>16</b></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> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1045.2 Field Documentation

8.1045.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceGetCOLR_t::pAlphaIDInfo`

8.1045.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetCOLR_t::ParamPresenceMask`

8.1045.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pCallID`

8.1045.2.4 `uint8_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pCCResType`

8.1045.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceGetCOLR_t::pCCSUPSType`

8.1045.2.6 `voice_COLRResp*` `unpack_voice_SLQSVoiceGetCOLR_t::pCOLRResp`

8.1045.2.7 `uint16_t*` `unpack_voice_SLQSVoiceGetCOLR_t::pFailCause`

## 8.1046 `unpack_voice_SLQSVoiceGetConfig_t` Struct Reference

### Data Fields

- `uint8_t *` `pAutoAnswerStat`
- `voice_airTimer *` `pAirTimerCnt`
- `voice_roamTimer *` `pRoamTimerCnt`

- uint8\_t \* pCurrTTYMode
- voice\_prefVoiceSO \* pCurPrefVoiceSO
- voice\_curAMRConfig \* pCurAMRConfig
- uint8\_t \* pCurVoicePrivacyPref
- uint8\_t \* pCurVoiceDomainPref
- swi\_uint256\_t ParamPresenceMask

### 8.1046.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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="#">voice_airTimer</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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="#">voice_roamTimer</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pCurPrefVoiceS-O(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="#">voice_prefVoiceSO</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>

<i>pCurAMR-Config(optional)</i>	<ul style="list-style-type: none"> <li>• Current Adaptive Multi-Rate Configuration.</li> <li>• Values returned are read from NV_GSM_ARM_CALL_CONFIG_I and NV_UMTS_AMR_CODEC_PREFERENCE_CONFIG_I.</li> <li>• See <a href="#">voice_curAMRConfig</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pCurVoice-Privacy-Pref(optional)</i>	<ul style="list-style-type: none"> <li>• Current Voice Privacy Preference</li> <li>• Value returned is read from NV_VOICE_PRIV_I. <ul style="list-style-type: none"> <li>– 0x00 - Standard privacy</li> <li>– 0x01 - Enhanced privacy</li> <li>– 0xFF - Not Available</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pCurVoice-Domain-Pref(optional)</i>	<ul style="list-style-type: none"> <li>• Current Voice Domain 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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1046.2 Field Documentation

8.1046.2.1 `voice_airTimer*` `unpack_voice_SLQSVoiceGetConfig_t::pAirTimerCnt`

8.1046.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceGetConfig_t::ParamPresenceMask`

8.1046.2.3 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pAutoAnswerStat`

8.1046.2.4 `voice_curAMRConfig*` `unpack_voice_SLQSVoiceGetConfig_t::pCurAMRConfig`

8.1046.2.5 `voice_prefVoiceSO*` `unpack_voice_SLQSVoiceGetConfig_t::pCurPrefVoiceSO`

8.1046.2.6 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurrTTYMode`

8.1046.2.7 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurVoiceDomainPref`

8.1046.2.8 `uint8_t*` `unpack_voice_SLQSVoiceGetConfig_t::pCurVoicePrivacyPref`

8.1046.2.9 `voice_roamTimer*` `unpack_voice_SLQSVoiceGetConfig_t::pRoamTimerCnt`

## 8.1047 `unpack_voice_SLQSVoiceManageCalls_t` Struct Reference

## Data Fields

- uint16\_t \* [pFailCause](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1047.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1047.2 Field Documentation

8.1047.2.1 [swi\\_uint256\\_t](#) [unpack\\_voice\\_SLQSVoiceManageCalls\\_t::ParamPresenceMask](#)

8.1047.2.2 [uint16\\_t\\*](#) [unpack\\_voice\\_SLQSVoiceManageCalls\\_t::pFailCause](#)

## 8.1048 unpack\_voice\_SLQSVoiceSendFlash\_t Struct Reference

## Data Fields

- uint8\_t \* [pCallID](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1048.1 Detailed Description

This structure contains the flash information associated with a call.

#### Parameters

<i>pCallID</i>	<ul style="list-style-type: none"> <li>• Unique call identifier associated with the current call.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1048.2 Field Documentation

8.1048.2.1 [swi\\_uint256\\_t](#) [unpack\\_voice\\_SLQSVoiceSendFlash\\_t::ParamPresenceMask](#)

8.1048.2.2 uint8\_t\* unpack\_voice\_SLQSVoiceSendFlash\_t::pCallID

## 8.1049 unpack\_voice\_SLQSVoiceSetCallBarringPassword\_t Struct Reference

### Data Fields

- uint16\_t \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- uint8\_t \* [pCCResType](#)
- uint8\_t \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1049.1 Detailed Description

This structure contains Voice Set Call Barring Password Unpack 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pAlphaIDInfo</i>	<ul style="list-style-type: none"> <li>• Pointer to structure of alphaIDInfo (optional) <ul style="list-style-type: none"> <li>– See <a href="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1049.2 Field Documentation

8.1049.2.1 `voice_alphalDInfo*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pAlphalDInfo`

8.1049.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::ParamPresenceMask`

8.1049.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCallID`

8.1049.2.4 `uint8_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCResType`

8.1049.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pCCSUPSType`

8.1049.2.6 `uint16_t*` `unpack_voice_SLQSVoiceSetCallBarringPassword_t::pFailCause`

## 8.1050 unpack\_voice\_SLQSVoiceSetConfig\_t Struct Reference

### Data Fields

- `uint8_t *` [pAutoAnsStatus](#)
- `uint8_t *` [pAirTimerStatus](#)
- `uint8_t *` [pRoamTimerStatus](#)
- `uint8_t *` [pTTYConfigStatus](#)
- `uint8_t *` [pPrefVoiceSOSStatus](#)
- `uint8_t *` [pVoiceDomainPrefStatus](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.1050.1 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>18</b></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> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1050.2 Field Documentation

8.1050.2.1 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAirTimerStatus`

8.1050.2.2 `swi_uint256_t unpack_voice_SLQSVoiceSetConfig_t::ParamPresenceMask`

8.1050.2.3 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pAutoAnsStatus`

8.1050.2.4 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pPrefVoiceSOSStatus`

8.1050.2.5 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pRoamTimerStatus`

8.1050.2.6 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pTTYConfigStatus`

8.1050.2.7 `uint8_t* unpack_voice_SLQSVoiceSetConfig_t::pVoiceDomainPrefStatus`



## 8.1051 unpack\_voice\_SLQSVoiceSetSUPSService\_t Struct Reference

### Data Fields

- uint16\_t \* [pFailCause](#)
- [voice\\_alphaIDInfo](#) \* [pAlphaIDInfo](#)
- uint8\_t \* [pCCResultType](#)
- uint8\_t \* [pCallID](#)
- [voice\\_ccSUPSType](#) \* [pCCSUPSType](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1051.1 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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></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="#">voice_alphaIDInfo</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pCCSUPSType</i>	<ul style="list-style-type: none"> <li>• Data is present when <a href="#">pCCResultType</a> is present and is other than Voice. (optional, supply NULL if not required) <ul style="list-style-type: none"> <li>– See <a href="#">voice_ccSUPSType</a> for more information</li> </ul> </li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>20</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1051.2 Field Documentation

8.1051.2.1 `voice_alphalDInfo*` `unpack_voice_SLQSVoiceSetSUPSService_t::pAlphalDInfo`

8.1051.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSetSUPSService_t::ParamPresenceMask`

8.1051.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCallID`

8.1051.2.4 `uint8_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCCResultType`

8.1051.2.5 `voice_ccSUPSType*` `unpack_voice_SLQSVoiceSetSUPSService_t::pCCSUPSType`

8.1051.2.6 `uint16_t*` `unpack_voice_SLQSVoiceSetSUPSService_t::pFailCause`

## 8.1052 `unpack_voice_SLQSVoiceStartContDTMF_t` Struct Reference

### Data Fields

- `uint8_t *` `pCallID`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1052.1 Detailed Description

This structure contains parameters of continuous DTMF

#### Parameters

<i>pCallID</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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1052.2 Field Documentation

8.1052.2.1 `swi_uint256_t` `unpack_voice_SLQSVoiceStartContDTMF_t::ParamPresenceMask`

8.1052.2.2 `uint8_t*` `unpack_voice_SLQSVoiceStartContDTMF_t::pCallID`

## 8.1053 `unpack_voice_SLQSVoiceStopContDTMF_t` Struct Reference

### Data Fields

- `uint8_t` `callID`
- `swi_uint256_t` `ParamPresenceMask`

### 8.1053.1 Detailed Description

This structure contains parameters of stop continuous DTMF

#### Parameters

<i>pCallID</i>	<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>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1053.2 Field Documentation

8.1053.2.1 `uint8_t unpack_voice_SLQSVoiceStopContDTMF_t::callID`

8.1053.2.2 `swi_uint256_t unpack_voice_SLQSVoiceStopContDTMF_t::ParamPresenceMask`

## 8.1054 unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t Struct Reference

#### Data Fields

- [voice\\_SUPSInfo](#) SUPSInformation
- `uint8_t * pSvcClass`
- `uint8_t * pReason`
- `uint8_t * pCallFWNum`
- `uint8_t * pCallFWTimerVal`
- `struct voice_USSInfo * pUSSInfo`
- `uint8_t * pCallID`
- `voice_alphaIDInfo * pAlphaIDInfo`
- `uint8_t * pCallBarPasswd`
- `voice_newPwdData * pNewPwdData`
- `uint8_t * pDataSrc`
- `uint16_t * pFailCause`
- `voice_getCallFWInfo * pCallFwdInfo`
- `voice_CLIRResp * pCLIRstatus`
- `voice_CLIPResp * pCLIPstatus`
- `voice_COLPResp * pCOLPstatus`
- `voice_COLRResp * pCOLRstatus`
- `voice_CNAPResp * pCNAPstatus`
- `swi_uint256_t ParamPresenceMask`

### 8.1054.1 Detailed Description

This structure contains the parameters passed for SUPS info indication by the device.

## Parameters

<i>SUPS- Information(mandatory)</i>	<ul style="list-style-type: none"> <li>See <a href="#">voice_SUPSInfo</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pSvc- Class(optional)</i>	<ul style="list-style-type: none"> <li>Service class is a combination (sum) of information class constants (optional)</li> <li>See <a href="#">qaGobiApiTableSupServiceInfoClasses.h</a> for service classes.</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>p- Reason(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTableCallControlReturnReasons.h</a> for return reasons.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></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> <li>Bit to check in ParamPresenceMask - <b>18</b></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> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>pUSS- Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">voice_USSInfo</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>20</b></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> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pAlphaID- Info(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">voice_alphaIDInfo</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>22</b></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> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pNewPwd- Data(optional)</i>	<ul style="list-style-type: none"> <li>See <a href="#">voice_newPwdData</a> for more information.</li> <li>Bit to check in ParamPresenceMask - <b>24</b></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> <li>• Bit to check in ParamPresenceMask - <b>25</b></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> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pCallFwd-Info(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_getCallFWInfo</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pCLI-Rstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_CLIRResp</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pCLI-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_CLIPResp</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pCOL-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_COLPResp</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pCOL-Rstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_COLRResp</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>pCNA-Pstatus(optional)</i>	<ul style="list-style-type: none"> <li>• See <a href="#">voice_CNAPResp</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## Note

None

## 8.1054.2 Field Documentation

8.1054.2.1 `voice_alphaIDInfo*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pAlphaIDInfo`8.1054.2.2 `swi_uint256_t` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::ParamPresenceMask`8.1054.2.3 `uint8_t*` `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallBarPasswd`

- 8.1054.2.4 **voice\_getCallFWInfo\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFwdInfo`
- 8.1054.2.5 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWNum`
- 8.1054.2.6 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallFWTimerVal`
- 8.1054.2.7 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCallIID`
- 8.1054.2.8 **voice\_CLIPResp\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIPstatus`
- 8.1054.2.9 **voice\_CLIRResp\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCLIRstatus`
- 8.1054.2.10 **voice\_CNAPResp\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCNAPstatus`
- 8.1054.2.11 **voice\_COLPResp\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLPstatus`
- 8.1054.2.12 **voice\_COLRResp\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pCOLRstatus`
- 8.1054.2.13 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pDataSrc`
- 8.1054.2.14 **uint16\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pFailCause`
- 8.1054.2.15 **voice\_newPwdData\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pNewPwdData`
- 8.1054.2.16 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pReason`
- 8.1054.2.17 **uint8\_t\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pSvcClass`
- 8.1054.2.18 **struct voice\_USSInfo\*** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::pUSSInfo`
- 8.1054.2.19 **voice\_SUPSInfo** `unpack_voice_SLQSVoiceSUPSCallback_ind_t::SUPSInformation`

## 8.1055 `unpack_voice_SUPSNotificationCallback_ind_t` Struct Reference

### Data Fields

- `uint8_t callIID`
- `uint8_t notifType`
- `uint16_t * pCUGIndex`
- `voice_ECTNum * pECTNum`
- `swi_uint256_t ParamPresenceMask`

### 8.1055.1 Detailed Description

Contains the parameters passed for SUPS notification indication by the device.

#### Parameters

<i>callIID</i>	<ul style="list-style-type: none"> <li>• Unique identifier of the call for which the notification is applicable. (mandatory)</li> <li>• Bit to check in ParamPresenceMask - 1</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> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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 ECTNum for details.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## Note

None

## 8.1055.2 Field Documentation

8.1055.2.1 `uint8_t unpack_voice_SUPSNotificationCallback_ind_t::callID`8.1055.2.2 `uint8_t unpack_voice_SUPSNotificationCallback_ind_t::notifType`8.1055.2.3 `swi_uint256_t unpack_voice_SUPSNotificationCallback_ind_t::ParamPresenceMask`8.1055.2.4 `uint16_t* unpack_voice_SUPSNotificationCallback_ind_t::pCUGIndex`8.1055.2.5 `voice_ECTNum* unpack_voice_SUPSNotificationCallback_ind_t::pECTNum`8.1056 `unpack_voice_USSDNotificationCallback_ind_t` Struct Reference

## Data Fields

- `uint8_t notification_Type`
- `voice_USSDNotificationNetworkInfo` [USSDNotificationNetworkInfo](#)
- `swi_uint256_t ParamPresenceMask`

## 8.1056.1 Detailed Description

Structure for storing the USSD notification indication parameters.

## Parameters

<i>notification_Type(mandatory)</i>	Values: -0x01-FURTHER_USER_ACTION_NOT_REQUIRED -0x02-FURTHER_USER_ACTION_REQUIRED • Bit to check in ParamPresenceMask - <b>1</b>
<i>USSD-Notification-Network-Info(optional)</i>	• USSD network info (See <a href="#">voice_USSDNotificationNetworkInfo</a> ) – Bit to check in ParamPresenceMask - <b>16</b>
<i>ParamPresence-Mask</i>	• bitmask representation to indicate valid parameters.

## 8.1056.2 Field Documentation

8.1056.2.1 `uint8_t unpack_voice_USSDNotificationCallback_ind_t::notification_Type`8.1056.2.2 `swi_uint256_t unpack_voice_USSDNotificationCallback_ind_t::ParamPresenceMask`8.1056.2.3 `voice_USSDNotificationNetworkInfo` `unpack_voice_USSDNotificationCallback_ind_t::USSDNotification-NetworkInfo`8.1057 `unpack_voice_VoiceInfoRecCallback_ind_t` Struct Reference



## Data Fields

- uint8\_t [callID](#)
- [voice\\_signalInfo](#) \* [pSignalInfo](#)
- [voice\\_callerIDInfo](#) \* [pCallerIDInfo](#)
- uint8\_t \* [pDispInfo](#)
- uint8\_t \* [pExtDispInfo](#)
- uint8\_t \* [pCallerNameInfo](#)
- uint8\_t \* [pCallWaitInd](#)
- [voice\\_connectNumInfo](#) \* [pConnectNumInfo](#)
- [voice\\_connectNumInfo](#) \* [pCallingPartyInfo](#)
- [voice\\_calledPartyInfo](#) \* [pCalledPartyInfo](#)
- [voice\\_redirNumInfo](#) \* [pRedirNumInfo](#)
- uint8\_t \* [pCLIRCause](#)
- [voice\\_NSSAudioCtrl](#) \* [pNSSAudioCtrl](#)
- uint8\_t \* [pNSSRelease](#)
- [voice\\_lineCtrlInfo](#) \* [pLineCtrlInfo](#)
- [voice\\_extDispRecInfo](#) \* [pExtDispRecInfo](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.1057.1 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> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>1</b></li> </ul>
<i>pSignalInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Signal Information</li> <li>• See <a href="#">voice_signalInfo</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>16</b></li> </ul>
<i>pCallerIDInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Caller ID Information</li> <li>• See <a href="#">voice_callerIDInfo</a> for more information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>17</b></li> </ul>
<i>pDispInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Display Information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>18</b></li> </ul>
<i>pExtDispInfo</i> [- Optional]	<ul style="list-style-type: none"> <li>• Extended Display Information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>19</b></li> </ul>
<i>pCallerName- Info</i> [Optional]	<ul style="list-style-type: none"> <li>• Caller Name Information</li> <li>• Bit to check in <a href="#">ParamPresenceMask</a> - <b>20</b></li> </ul>

<i>pCallWaitInd[-Optional]</i>	<ul style="list-style-type: none"> <li>• Call Waiting Indicator</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pConnectNumInfo[Optional]</i>	<ul style="list-style-type: none"> <li>• Connected Number Information</li> <li>• see <a href="#">voice_connectNumInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>pCallingPartyInfo[Optional]</i>	<ul style="list-style-type: none"> <li>• Calling Party Number Information</li> <li>• This structure is having exactly same elements as connectNumInfo</li> <li>• see <a href="#">voice_connectNumInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>pCalledPartyInfo[Optional]</i>	<ul style="list-style-type: none"> <li>• Called Party Number Information</li> <li>• see <a href="#">voice_calledPartyInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>pRedirNumInfo[-Optional]</i>	<ul style="list-style-type: none"> <li>• Redirecting Number Information</li> <li>• see <a href="#">voice_redirNumInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pCLIRCause[-Optional]</i>	<ul style="list-style-type: none"> <li>• National Supplementary Services - CLIR</li> <li>• see <a href="#">voice_NSSAudioCtrl</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pNSSAudioCtrl[-Optional]</i>	<ul style="list-style-type: none"> <li>• National Supplementary Services - Audio Control</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pNSSRelease[-Optional]</i>	<ul style="list-style-type: none"> <li>• National Supplementary Services - Release</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>pLineCtrlInfo[-Optional]</i>	<ul style="list-style-type: none"> <li>• Line Control Information</li> <li>• see <a href="#">voice_lineCtrlInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pExtDispRecInfo[Optional]</i>	<ul style="list-style-type: none"> <li>• Extended Display Record Information</li> <li>• see <a href="#">voice_extDispRecInfo</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1057.2 Field Documentation

- 8.1057.2.1 `uint8_t unpack_voice_VoiceInfoRecCallback_ind_t::callID`
- 8.1057.2.2 `swi_uint256_t unpack_voice_VoiceInfoRecCallback_ind_t::ParamPresenceMask`
- 8.1057.2.3 `voice_calledPartyInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCalledPartyInfo`
- 8.1057.2.4 `voice_callerIDInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerIDInfo`
- 8.1057.2.5 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallerNameInfo`
- 8.1057.2.6 `voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pCallingPartyInfo`
- 8.1057.2.7 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCallWaitInd`
- 8.1057.2.8 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pCLIRCause`
- 8.1057.2.9 `voice_connectNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pConnectNumInfo`
- 8.1057.2.10 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pDisplInfo`
- 8.1057.2.11 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDisplInfo`
- 8.1057.2.12 `voice_extDispRecInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pExtDispRecInfo`
- 8.1057.2.13 `voice_lineCtrlInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pLineCtrlInfo`
- 8.1057.2.14 `voice_NSSAudioCtrl* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSAudioCtrl`
- 8.1057.2.15 `uint8_t* unpack_voice_VoiceInfoRecCallback_ind_t::pNSSRelease`
- 8.1057.2.16 `voice_redirNumInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pRedirNumInfo`
- 8.1057.2.17 `voice_signalInfo* unpack_voice_VoiceInfoRecCallback_ind_t::pSignalInfo`

## 8.1058 unpack\_voice\_voicePrivacyChangeCallback\_ind\_t Struct Reference

### Data Fields

- `uint8_t callID`
- `uint8_t voicePrivacy`
- `swi_uint256_t ParamPresenceMask`

### 8.1058.1 Detailed Description

Contains the parameters passed for voice privacy change indication

#### 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><li>• Bit to check in ParamPresenceMask - 1</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> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

#### Note

None

### 8.1058.2 Field Documentation

8.1058.2.1 `uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::callID`

8.1058.2.2 `swi_uint256_t unpack_voice_voicePrivacyChangeCallback_ind_t::ParamPresenceMask`

8.1058.2.3 `uint8_t unpack_voice_voicePrivacyChangeCallback_ind_t::voicePrivacy`

### 8.1059 `unpack_wds_DHCPv4ClientLease_ind_t` Struct Reference

#### Data Fields

- [wds\\_DHCPProfileIdTlv ProfileIdTlv](#)
- [wds\\_DHCPLeaseStateTlv DHCPv4LeaseStateTlv](#)
- [wds\\_IPv4AddrTlv IPv4AddrTlv](#)
- [wds\\_DHCPLeaseOptTlv DHCPv4LeaseOptTlv](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

#### 8.1059.1 Detailed Description

This structure contains DHCP IPv4 client lease information

##### Parameters

<i>ProfileIdTlv</i>	profile identifier information, see <a href="#">wds_DHCPProfileIdTlv</a> for more details <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>DHCPv4LeaseStateTlv</i>	lease state, see <a href="#">wds_DHCPLeaseStateTlv</a> for more details <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>2</b></li> </ul>
<i>IPv4AddrTlv</i>	IPv4 address, see <a href="#">wds_IPv4AddrTlv</a> for more details <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>DHCPv4LeaseOptTlv</i>	DHCP lease option, see <a href="#">wds_DHCPLeaseOptTlv</a> for more details <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1059.2 Field Documentation

8.1059.2.1 `wds_DHCPLeaseOptTlv` `unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseOptTlv`

8.1059.2.2 `wds_DHCPLeaseStateTlv` `unpack_wds_DHCPv4ClientLease_ind_t::DHCPv4LeaseStateTlv`

8.1059.2.3 `wds_IPv4AdTlv` `unpack_wds_DHCPv4ClientLease_ind_t::IPv4AddrTlv`

8.1059.2.4 `swi_uint256_t` `unpack_wds_DHCPv4ClientLease_ind_t::ParamPresenceMask`

8.1059.2.5 `wds_DHCPProfileIdTlv` `unpack_wds_DHCPv4ClientLease_ind_t::ProfileIdTlv`

## 8.1060 unpack\_wds\_GetAutoconnect\_t Struct Reference

### Data Fields

- `uint32_t * psetting`
- `swi_uint256_t ParamPresenceMask`

### 8.1060.1 Detailed Description

auto connect data session setting parameter.

#### Parameters

<i>pSetting</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1060.2 Field Documentation

8.1060.2.1 `swi_uint256_t` `unpack_wds_GetAutoconnect_t::ParamPresenceMask`

8.1060.2.2 `uint32_t*` `unpack_wds_GetAutoconnect_t::psetting`

## 8.1061 unpack\_wds\_GetByteTotals\_t Struct Reference

### Data Fields

- `uint64_t * pTXTotalBytes`
- `uint64_t * pRXTotalBytes`
- `swi_uint256_t ParamPresenceMask`

### 8.1061.1 Detailed Description

This structure contains unpack get byte totals information.

## Parameters

<i>pTXTotalBytes</i>	<ul style="list-style-type: none"> <li>Bytes transmitted without error</li> <li>Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pRXTotalBytes</i>	<ul style="list-style-type: none"> <li>Bytes received without error</li> <li>Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1061.2 Field Documentation

8.1061.2.1 `swi_uint256_t unpack_wds_GetByteTotals_t::ParamPresenceMask`8.1061.2.2 `uint64_t* unpack_wds_GetByteTotals_t::pRXTotalBytes`8.1061.2.3 `uint64_t* unpack_wds_GetByteTotals_t::pTXTotalBytes`8.1062 `unpack_wds_GetConnectionRate_t` Struct Reference

## Data Fields

- `uint32_t` [currentChannelTXRate](#)
- `uint32_t` [currentChannelRXRate](#)
- `uint32_t` [maxChannelTXRate](#)
- `uint32_t` [maxChannelRXRate](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.1062.1 Detailed Description

This structure contains unpack get connection rate information.

## Parameters

<i>currentChannelTXRate</i>	<ul style="list-style-type: none"> <li>Current channel Tx rate (in bps)</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>currentChannelRXRate</i>	<ul style="list-style-type: none"> <li>Current channel Rx rate (in bps)</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>maxChannelTXRate</i>	<ul style="list-style-type: none"> <li>Maximum Tx rate (bps) that may be assigned to device by serving system.</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>maxChannelRXRate</i>	<ul style="list-style-type: none"> <li>Maximum Rx rate (bps) that may be assigned to device by serving system.</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"><li>• bitmask representation to indicate valid parameters.</li></ul>
--------------------------	--

## 8.1062.2 Field Documentation

8.1062.2.1 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelRXRate`

8.1062.2.2 `uint32_t unpack_wds_GetConnectionRate_t::currentChannelTXRate`

8.1062.2.3 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelRXRate`

8.1062.2.4 `uint32_t unpack_wds_GetConnectionRate_t::maxChannelTXRate`

8.1062.2.5 `swi_uint256_t unpack_wds_GetConnectionRate_t::ParamPresenceMask`

## 8.1063 unpack\_wds\_GetDataBearerTechnology\_t Struct Reference

### Data Fields

- `uint32_t *` [pDataBearer](#)
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.1063.1 Detailed Description

This structure contains unpack get data bearer technology information.

#### Parameters

---

<i>pDataBearer[O-UT]</i>	<ul style="list-style-type: none"> <li>• Data bearer technology <ul style="list-style-type: none"> <li>– 0x01 - CDMA2000 1x</li> <li>– 0x02 - CDMA 1xEV-DO Rev 0</li> <li>– 0x03 - GSM</li> <li>– 0x04 - UMTS</li> <li>– 0x05 - CDMA2000 HRPD (1xEV-DO Rev A)</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA AND WCDMA</li> <li>– 0x08 - WCDMA AND HSUPA</li> <li>– 0x09 - HSDPA AND HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - CDMA2000 EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x10 - HSDPA+ and 64QAM</li> <li>– 0x11 - HSDPA+, 64QAM and HSUPA</li> <li>– 0x12 - TDSCDMA</li> <li>– 0x13 - TDSCDMA and HSDPA</li> <li>– 0xFF - Unknown</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1063.2 Field Documentation

8.1063.2.1 `swi_uint256_t unpack_wds_GetDataBearerTechnology_t::ParamPresenceMask`

8.1063.2.2 `uint32_t* unpack_wds_GetDataBearerTechnology_t::pDataBearer`

## 8.1064 `unpack_wds_GetDefaultProfile_t` Struct Reference

### Data Fields

- `uint32_t pdptype`
- `uint32_t ipaddr`
- `uint32_t pridns`
- `uint32_t secdns`
- `uint16_t ipaddrv6 [8]`
- `uint16_t pridnsv6 [8]`
- `uint16_t secdnsv6 [8]`
- `uint32_t auth`
- `uint8_t namesize`
- `int8_t name [255]`
- `uint8_t apnsiz`
- `int8_t apnname [255]`



- [uint8\\_t usersize](#)
- [int8\\_t username](#) [255]
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1064.1 Detailed Description

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

#### Parameters

<i>pdptype</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> <li>– 1 - PDP-PPP</li> <li>– 2 - PDP-IPv6</li> <li>– 3 - PDP-IPv4v6</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ipaddr</i>	<ul style="list-style-type: none"> <li>• Preferred IPv4 address to be assigned to device</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pridns</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>secdns</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ipaddrv6</i>	<ul style="list-style-type: none"> <li>• Preferred IPv6 address to be assigned to device</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>
<i>pridnsv6</i>	<ul style="list-style-type: none"> <li>• Primary DNS Pv6 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>
<i>secdnsv6</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>

<i>auth</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> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>name</i>	<ul style="list-style-type: none"> <li>• Profile name</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>apnname</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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>username</i>	<ul style="list-style-type: none"> <li>• Username used during network authentication</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1064.2 Field Documentation

8.1064.2.1 `int8_t unpack_wds_GetDefaultProfile_t::apnname[255]`

8.1064.2.2 `uint8_t unpack_wds_GetDefaultProfile_t::apnsize`

8.1064.2.3 `uint32_t unpack_wds_GetDefaultProfile_t::auth`

- 8.1064.2.4 `uint32_t unpack_wds_GetDefaultProfile_t::ipaddr`
- 8.1064.2.5 `uint16_t unpack_wds_GetDefaultProfile_t::ipaddrv6[8]`
- 8.1064.2.6 `int8_t unpack_wds_GetDefaultProfile_t::name[255]`
- 8.1064.2.7 `uint8_t unpack_wds_GetDefaultProfile_t::namesize`
- 8.1064.2.8 `swi_uint256_t unpack_wds_GetDefaultProfile_t::ParamPresenceMask`
- 8.1064.2.9 `uint32_t unpack_wds_GetDefaultProfile_t::pdptype`
- 8.1064.2.10 `uint32_t unpack_wds_GetDefaultProfile_t::pridns`
- 8.1064.2.11 `uint16_t unpack_wds_GetDefaultProfile_t::pridnsv6[8]`
- 8.1064.2.12 `uint32_t unpack_wds_GetDefaultProfile_t::secdns`
- 8.1064.2.13 `uint16_t unpack_wds_GetDefaultProfile_t::secdnsv6[8]`
- 8.1064.2.14 `int8_t unpack_wds_GetDefaultProfile_t::username[255]`
- 8.1064.2.15 `uint8_t unpack_wds_GetDefaultProfile_t::usersize`

## 8.1065 unpack\_wds\_GetDefaultProfileNum\_t Struct Reference

### Data Fields

- [uint8\\_t index](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1065.1 Detailed Description

This structure contains unpack Get Default Profile Number information.

#### Parameters

<i>index</i>	profile index <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1065.2 Field Documentation

- 8.1065.2.1 `uint8_t unpack_wds_GetDefaultProfileNum_t::index`
- 8.1065.2.2 `swi_uint256_t unpack_wds_GetDefaultProfileNum_t::ParamPresenceMask`

## 8.1066 unpack\_wds\_GetDefaultProfileV2\_t Struct Reference

## Data Fields

- uint32\_t [pdptype](#)
- uint32\_t [ipaddr](#)
- uint32\_t [pridns](#)
- uint32\_t [secdns](#)
- uint16\_t [ipaddrv6](#) [8]
- uint16\_t [pridnsv6](#) [8]
- uint16\_t [secdnsv6](#) [8]
- uint32\_t [auth](#)
- uint8\_t [namesize](#)
- int8\_t [name](#) [255]
- uint8\_t [apnsize](#)
- int8\_t [apnname](#) [255]
- uint8\_t [usersize](#)
- int8\_t [username](#) [255]
- uint8\_t [pwdsize](#)
- int8\_t [pwd](#) [255]
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1066.1 Detailed Description

Reads the default profile settings from the device. The default profile is used to establish an auto connect data session.

#### Parameters

<i>pdptype</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> <li>– 1 - PDP-PPP</li> <li>– 2 - PDP-IPv6</li> <li>– 3 - PDP-IPv4v6</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ipaddr</i>	<ul style="list-style-type: none"> <li>• Preferred IPv4 address to be assigned to device</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>pridns</i>	<ul style="list-style-type: none"> <li>• Primary DNS IPv4 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>secdns</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv4 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ipaddrv6</i>	<ul style="list-style-type: none"> <li>• Preferred IPv6 address to be assigned to device</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>

<i>pridnsv6</i>	<ul style="list-style-type: none"> <li>• Primary DNS Pv6 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>
<i>secdnsv6</i>	<ul style="list-style-type: none"> <li>• Secondary DNS IPv6 address preference</li> <li>• Bit to check in ParamPresenceMask - <b>143</b></li> </ul>
<i>auth</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> <li>• Bit to check in ParamPresenceMask - <b>29</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>name</i>	<ul style="list-style-type: none"> <li>• Profile name</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>apnname</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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>username</i>	<ul style="list-style-type: none"> <li>• Username used during network authentication</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>pwdsize</i>	<ul style="list-style-type: none"> <li>• Maximum number of characters (including NULL terminator) that pwd name array can contain.</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>

<i>pwd</i>	<ul style="list-style-type: none"> <li>• Password used during network authentication</li> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1066.2 Field Documentation

- 8.1066.2.1 `int8_t unpack_wds_GetDefaultProfileV2_t::apnname[255]`
- 8.1066.2.2 `uint8_t unpack_wds_GetDefaultProfileV2_t::apnsize`
- 8.1066.2.3 `uint32_t unpack_wds_GetDefaultProfileV2_t::auth`
- 8.1066.2.4 `uint32_t unpack_wds_GetDefaultProfileV2_t::ipaddr`
- 8.1066.2.5 `uint16_t unpack_wds_GetDefaultProfileV2_t::ipaddrv6[8]`
- 8.1066.2.6 `int8_t unpack_wds_GetDefaultProfileV2_t::name[255]`
- 8.1066.2.7 `uint8_t unpack_wds_GetDefaultProfileV2_t::namesize`
- 8.1066.2.8 `swi_uint256_t unpack_wds_GetDefaultProfileV2_t::ParamPresenceMask`
- 8.1066.2.9 `uint32_t unpack_wds_GetDefaultProfileV2_t::pdptype`
- 8.1066.2.10 `uint32_t unpack_wds_GetDefaultProfileV2_t::pridns`
- 8.1066.2.11 `uint16_t unpack_wds_GetDefaultProfileV2_t::pridnsv6[8]`
- 8.1066.2.12 `int8_t unpack_wds_GetDefaultProfileV2_t::pwd[255]`
- 8.1066.2.13 `uint8_t unpack_wds_GetDefaultProfileV2_t::pwdsize`
- 8.1066.2.14 `uint32_t unpack_wds_GetDefaultProfileV2_t::secdns`
- 8.1066.2.15 `uint16_t unpack_wds_GetDefaultProfileV2_t::secdnsv6[8]`
- 8.1066.2.16 `int8_t unpack_wds_GetDefaultProfileV2_t::username[255]`
- 8.1066.2.17 `uint8_t unpack_wds_GetDefaultProfileV2_t::usersize`

## 8.1067 `unpack_wds_GetDormancyState_t` Struct Reference

### Data Fields

- `uint32_t dormancyState`
- `swi_uint256_t ParamPresenceMask`

### 8.1067.1 Detailed Description

This structure contains unpack get dormancy state information.

#### Parameters

<i>dormancyState</i>	<ul style="list-style-type: none"> <li>Dormancy state of current packet data session <ul style="list-style-type: none"> <li>1 - Traffic channel dormant</li> <li>2 - Traffic channel active</li> </ul> </li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1067.2 Field Documentation

8.1067.2.1 `uint32_t unpack_wds_GetDormancyState_t::dormancyState`

8.1067.2.2 `swi_uint256_t unpack_wds_GetDormancyState_t::ParamPresenceMask`

## 8.1068 unpack\_wds\_GetLastMobileIPError\_t Struct Reference

### Data Fields

- `uint32_t error`
- `swi_uint256_t ParamPresenceMask`

### 8.1068.1 Detailed Description

structure to store last mobile IP error.

#### Parameters

<i>error</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> <li>See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Mobile IP Error codes</li> <li>Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

### 8.1068.2 Field Documentation

8.1068.2.1 `uint32_t unpack_wds_GetLastMobileIPError_t::error`

8.1068.2.2 `swi_uint256_t unpack_wds_GetLastMobileIPError_t::ParamPresenceMask`

## 8.1069 unpack\_wds\_GetMobileIP\_t Struct Reference

### Data Fields

- [uint32\\_t mipMode](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1069.1 Detailed Description

This structure contains unpack get mobile IP information.

#### Parameters

<i>mipMode</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> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1069.2 Field Documentation

8.1069.2.1 [uint32\\_t unpack\\_wds\\_GetMobileIP\\_t::mipMode](#)

8.1069.2.2 [swi\\_uint256\\_t unpack\\_wds\\_GetMobileIP\\_t::ParamPresenceMask](#)

## 8.1070 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 HAState](#)
- [uint32\\_t AAASState](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1070.1 Detailed Description

This structure contains unpack get mobile IP profile information.



## Parameters

<i>enabled</i>	<ul style="list-style-type: none"> <li>• Profile enabled: <ul style="list-style-type: none"> <li>– 0 - Disabled</li> <li>– 1 - Enabled</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>address</i>	<ul style="list-style-type: none"> <li>• Home IPv4 address:</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>primaryHA</i>	<ul style="list-style-type: none"> <li>• Primary home agent IPv4 address</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>secondaryHA</i>	<ul style="list-style-type: none"> <li>• Secondary home agent IPv4 address</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>revTunneling</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>NAI</i>	<ul style="list-style-type: none"> <li>• Network access identifier string</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>HASPI</i>	<ul style="list-style-type: none"> <li>• Home agent security parameter index</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>AAASPI</i>	<ul style="list-style-type: none"> <li>• AAA server security parameter index</li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>HASState</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> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>

<i>AAASState</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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1070.2 Field Documentation

- 8.1070.2.1 `uint32_t unpack_wds_GetMobileIPProfile_t::AAASPI`
- 8.1070.2.2 `uint32_t unpack_wds_GetMobileIPProfile_t::AAASState`
- 8.1070.2.3 `uint32_t unpack_wds_GetMobileIPProfile_t::address`
- 8.1070.2.4 `uint8_t unpack_wds_GetMobileIPProfile_t::enabled`
- 8.1070.2.5 `uint32_t unpack_wds_GetMobileIPProfile_t::HASPI`
- 8.1070.2.6 `uint32_t unpack_wds_GetMobileIPProfile_t::HASState`
- 8.1070.2.7 `int8_t unpack_wds_GetMobileIPProfile_t::NAI[255]`
- 8.1070.2.8 `uint8_t unpack_wds_GetMobileIPProfile_t::naiSize`
- 8.1070.2.9 `swi_uint256_t unpack_wds_GetMobileIPProfile_t::ParamPresenceMask`
- 8.1070.2.10 `uint32_t unpack_wds_GetMobileIPProfile_t::primaryHA`
- 8.1070.2.11 `uint8_t unpack_wds_GetMobileIPProfile_t::revTunneling`
- 8.1070.2.12 `uint32_t unpack_wds_GetMobileIPProfile_t::secondaryHA`

### 8.1071 `unpack_wds_GetPacketStatistics_t` Struct Reference

#### Data Fields

- `uint32_t * pTXPacketSuccesses`
- `uint32_t * pRXPacketSuccesses`
- `uint32_t * pTXPacketErrors`
- `uint32_t * pRXPacketErrors`
- `uint32_t * pTXPacketOverflows`
- `uint32_t * pRXPacketOverflows`
- `uint64_t * pTXOkBytesCount`
- `uint64_t * pRXOkBytesCount`
- `uint64_t * pTXOKBytesLastCall`
- `uint64_t * pRXOKBytesLastCall`
- `uint32_t * pTXDroppedCount`

- uint32\_t \* pRXDroppedCount
- swi\_uint256\_t ParamPresenceMask

### 8.1071.1 Detailed Description

This structure contains unpack get packet statistics information.

#### Parameters

<i>pTXPacket-Successes</i>	<ul style="list-style-type: none"> <li>• No of transmitted Packets without error.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pRXPacket-Successes</i>	<ul style="list-style-type: none"> <li>• No of received Packets without error.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pTXPacketErrors</i>	<ul style="list-style-type: none"> <li>• Number of outgoing packets with framing errors.</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>pRXPacket-Errors</i>	<ul style="list-style-type: none"> <li>• Number of incoming packets with framing errors.</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>pTXOkBytes-Count</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>pRXOkBytes-Count</i>	<ul style="list-style-type: none"> <li>• Number of bytes received without error.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>pTXOKBytes-LastCall</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 earlier). Returned only if not in a call, and when the previous call was made using RmNet (for any devices that support</li> <li>• Bit to check in ParamPresenceMask - <b>27</b></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> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>

<i>pTXDropped-Count</i>	<ul style="list-style-type: none"> <li>• Number of outgoing packets dropped.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>pRXDropped-Count</i>	<ul style="list-style-type: none"> <li>• Number of incoming packets dropped.</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1071.2 Field Documentation

8.1071.2.1 `swi_uint256_t unpack_wds_GetPacketStatistics_t::ParamPresenceMask`

8.1071.2.2 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXDroppedCount`

8.1071.2.3 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesCount`

8.1071.2.4 `uint64_t* unpack_wds_GetPacketStatistics_t::pRXOkBytesLastCall`

8.1071.2.5 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketErrors`

8.1071.2.6 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketOverflows`

8.1071.2.7 `uint32_t* unpack_wds_GetPacketStatistics_t::pRXPacketSuccesses`

8.1071.2.8 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXDroppedCount`

8.1071.2.9 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesCount`

8.1071.2.10 `uint64_t* unpack_wds_GetPacketStatistics_t::pTXOkBytesLastCall`

8.1071.2.11 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketErrors`

8.1071.2.12 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketOverflows`

8.1071.2.13 `uint32_t* unpack_wds_GetPacketStatistics_t::pTXPacketSuccesses`

## 8.1072 `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
- swi\_uint256\_t ParamPresenceMask

### 8.1072.1 Detailed Description

Returns the packet data transfer statistics since the start of the current packet data.

#### Parameters

<i>txPacket-Successes</i>	<ul style="list-style-type: none"> <li>• No. of packets transmitted without error</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>rxPacket-Successes</i>	<ul style="list-style-type: none"> <li>• No. of packets received without error</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>txPacketErrors</i>	<ul style="list-style-type: none"> <li>• No. of outgoing packets with framing errors</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>rxPacketErrors</i>	<ul style="list-style-type: none"> <li>• No. of incoming packets with framing errors</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>txPacket-Overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Tx buffer overflowed</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>rxPacket-Overflows</i>	<ul style="list-style-type: none"> <li>• Number of packets dropped because Rx buffer overflowed</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>txOkBytesCount</i>	<ul style="list-style-type: none"> <li>• No of bytes transmitted without error.</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>rxOkBytesCount</i>	<ul style="list-style-type: none"> <li>• No of bytes received without error.</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>txOKBytesLast-Call</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> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>rxOKBytesLast-Call</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> <li>• Bit to check in ParamPresenceMask - <b>28</b></li> </ul>

<i>txDroppedCount</i>	<ul style="list-style-type: none"> <li>• Number of outgoing packets dropped.</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>rxDroppedCount</i>	<ul style="list-style-type: none"> <li>• Number of incoming packets dropped.</li> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1072.2 Field Documentation

8.1072.2.1 `swi_uint256_t unpack_wds_GetPacketStatus_t::ParamPresenceMask`

8.1072.2.2 `uint32_t unpack_wds_GetPacketStatus_t::rXDroppedCount`

8.1072.2.3 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesCount`

8.1072.2.4 `uint64_t unpack_wds_GetPacketStatus_t::rXOkBytesLastCall`

8.1072.2.5 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketErrors`

8.1072.2.6 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketOverflows`

8.1072.2.7 `uint32_t unpack_wds_GetPacketStatus_t::rXPacketSuccesses`

8.1072.2.8 `uint32_t unpack_wds_GetPacketStatus_t::tXDroppedCount`

8.1072.2.9 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesCount`

8.1072.2.10 `uint64_t unpack_wds_GetPacketStatus_t::tXOkBytesLastCall`

8.1072.2.11 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketErrors`

8.1072.2.12 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketOverflows`

8.1072.2.13 `uint32_t unpack_wds_GetPacketStatus_t::tXPacketSuccesses`

## 8.1073 `unpack_wds_GetSessionDuration_t` Struct Reference

### Data Fields

- `uint64_t callDuration`
- `swi_uint256_t ParamPresenceMask`

### 8.1073.1 Detailed Description

This structure contains unpack get session duration information.

## Parameters

<i>callDuration</i>	<ul style="list-style-type: none"> <li>Duration of the current packet session in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1073.2 Field Documentation

8.1073.2.1 uint64\_t unpack\_wds\_GetSessionDuration\_t::callDuration

8.1073.2.2 swi\_uint256\_t unpack\_wds\_GetSessionDuration\_t::ParamPresenceMask

## 8.1074 unpack\_wds\_GetSessionDurationV2\_t Struct Reference

## Data Fields

- uint64\_t [callDuration](#)
- uint64\_t \* [pLastCallDuration](#)
- uint64\_t \* [pCallActiveDuration](#)
- uint64\_t \* [pLastCallActiveDuration](#)
- swi\_uint256\_t [ParamPresenceMask](#)

## 8.1074.1 Detailed Description

This structure contains unpack get session duration information -V2.

## Parameters

<i>callDuration</i>	<ul style="list-style-type: none"> <li>Duration of the current packet session in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>pLastCallDuration</i>	<ul style="list-style-type: none"> <li>Duration of the last data session in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pCallActiveDuration</i>	<ul style="list-style-type: none"> <li>Duration of the active time of current data session in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>pLastCallActiveDuration</i>	<ul style="list-style-type: none"> <li>Duration of the active time of last data session in milliseconds</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1074.2 Field Documentation

8.1074.2.1 `uint64_t unpack_wds_GetSessionDurationV2_t::callDuration`

8.1074.2.2 `swi_uint256_t unpack_wds_GetSessionDurationV2_t::ParamPresenceMask`

8.1074.2.3 `uint64_t* unpack_wds_GetSessionDurationV2_t::pCallActiveDuration`

8.1074.2.4 `uint64_t* unpack_wds_GetSessionDurationV2_t::pLastCallActiveDuration`

8.1074.2.5 `uint64_t* unpack_wds_GetSessionDurationV2_t::pLastCallDuration`

## 8.1075 `unpack_wds_GetSessionState_t` Struct Reference

### Data Fields

- `uint32_t connectionStatus`
- `swi_uint256_t ParamPresenceMask`

### 8.1075.1 Detailed Description

This structure contains unpack get session state information.

#### Parameters

<i><code>connection-Status</code></i>	- state of the current packet data session. <ul style="list-style-type: none"> <li>• 1 - Disconnected.</li> <li>• 2 - Connected.</li> <li>• 3 - Suspended.</li> <li>• 4 - Authenticating.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i><code>ParamPresence-Mask</code></i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1075.2 Field Documentation

8.1075.2.1 `uint32_t unpack_wds_GetSessionState_t::connectionStatus`

8.1075.2.2 `swi_uint256_t unpack_wds_GetSessionState_t::ParamPresenceMask`

## 8.1076 `unpack_wds_RMSetTransferStatistics_t` Struct Reference

### Data Fields

- `swi_uint256_t ParamPresenceMask`

### 8.1076.1 Detailed Description

This structure contains unpack fetch current data system transfer statistics information.



## Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.1076.2 Field Documentation

8.1076.2.1 `swi_uint256_t unpack_wds_RMSetTransferStatistics_t::ParamPresenceMask`

## 8.1077 unpack\_wds\_SetMobileIPProfile\_t Struct Reference

## Data Fields

- [swi\\_uint256\\_t ParamPresenceMask](#)

## 8.1077.1 Detailed Description

This structure contains set mobile IP profile information.

## Parameters

<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>
--------------------------	--

## 8.1077.2 Field Documentation

8.1077.2.1 `swi_uint256_t unpack_wds_SetMobileIPProfile_t::ParamPresenceMask`

## 8.1078 unpack\_wds\_SLQSCreateProfile\_t Struct Reference

## Data Fields

- [PackCreateProfileOut \\* pCreateProfileOut](#)
- `uint8_t * pProfileID`
- `uint16_t Tlvresult`
- [swi\\_uint256\\_t ParamPresenceMask](#)

## 8.1078.1 Detailed Description

This structure contains unpack create profile information.

## Parameters

<i>pCreateProfileOut</i>	<ul style="list-style-type: none"> <li>SLQS Create profile Information</li> <li>See <a href="#">PackCreateProfileOut</a></li> </ul>
<i>pProfileID</i>	<ul style="list-style-type: none"> <li>SLQS profile identifier information</li> </ul>

<i>Tlvresult</i>	<ul style="list-style-type: none"> <li>• TLV present flag, TRUE if presented, otherwise FALSE</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1078.2 Field Documentation

8.1078.2.1 `swi_uint256_t unpack_wds_SLQSCreateProfile_t::ParamPresenceMask`

8.1078.2.2 `PackCreateProfileOut* unpack_wds_SLQSCreateProfile_t::pCreateProfileOut`

8.1078.2.3 `uint8_t* unpack_wds_SLQSCreateProfile_t::pProfileID`

8.1078.2.4 `uint16_t unpack_wds_SLQSCreateProfile_t::Tlvresult`

## 8.1079 unpack\_wds\_SLQSDeleteProfile\_t Struct Reference

### Data Fields

- `uint16_t` [extendedErrorCode](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.1079.1 Detailed Description

This structure contains unpack delete profile information.

#### Parameters

<i>extendedErrorCode[OUT]</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> <li>• Bit to check in <code>ParamPresenceMask</code> - <b>224</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1079.2 Field Documentation

8.1079.2.1 `uint16_t unpack_wds_SLQSDeleteProfile_t::extendedErrorCode`

8.1079.2.2 `swi_uint256_t unpack_wds_SLQSDeleteProfile_t::ParamPresenceMask`

## 8.1080 unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t Struct Reference

## Data Fields

- [wds\\_ConnStatusTlv](#) CSTlv
- [wds\\_LastMdmCallEndRsnTlv](#) LMCERTlv
- [wds\\_TXBytesOKTlv](#) TXBOTlv
- [wds\\_RXBytesOKTlv](#) RXBOTlv
- [wds\\_DormStatTlv](#) DSTlv
- [wds\\_DataBearTechTlv](#) DBTTlv
- [wds\\_ChannelRateTlv](#) CRTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

## 8.1080.1 Detailed Description

Structure used to store Reg Mgr Config Indication Parameters.

## Parameters

<i>CSTlv</i>	<ul style="list-style-type: none"> <li>• Connection status tlv</li> <li>• See <a href="#">wds_ConnStatusTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>LMCERTlv</i>	<ul style="list-style-type: none"> <li>• last modem call end reason tlv</li> <li>• see <a href="#">wds_LastMdmCallEndRsnTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>TXBOTlv</i>	<ul style="list-style-type: none"> <li>• Tx bytes OK tlv</li> <li>• see <a href="#">wds_TXBytesOKTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>RXBOTlv</i>	<ul style="list-style-type: none"> <li>• Rx bytes OK tlv</li> <li>• see <a href="#">wds_RXBytesOKTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>DSTlv</i>	<ul style="list-style-type: none"> <li>• Dormancy status tlv</li> <li>• see <a href="#">wds_DormStatTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>DBTTlv</i>	<ul style="list-style-type: none"> <li>• Data bear technology tlv</li> <li>• see <a href="#">wds_DataBearTechTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>CRTlv</i>	<ul style="list-style-type: none"> <li>• Channel rate Tlv</li> <li>• see <a href="#">wds_ChannelRateTlv</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## Note

: None

## 8.1080.2 Field Documentation

8.1080.2.1 `wds_ChannelRateTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CRTlv`8.1080.2.2 `wds_ConnStatusTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::CSTlv`8.1080.2.3 `wds_DataBearTechTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DBTTlv`8.1080.2.4 `wds_DormStatTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::DSTlv`8.1080.2.5 `wds_LastMdmCallEndRsnTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::LMCERTlv`8.1080.2.6 `swi_uint256_t` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::ParamPresenceMask`8.1080.2.7 `wds_RXBytesOKTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::RXBOTlv`8.1080.2.8 `wds_TXBytesOKTlv` `unpack_wds_SLQSDUNCallInfoCallBack_ind_t::TXBOTlv`8.1081 `unpack_wds_SLQSGet3GPPConfigItem_t` Struct Reference

## Data Fields

- `uint16_t` [LTEAttachProfile](#)
- `uint16_t` [profileList](#) [5]
- `uint8_t` [defaultPDNEnabled](#)
- `uint8_t` [\\_3gppRelease](#)
- `uint16_t` [LTEAttachProfileList](#) [24]
- `uint16_t` [LTEAttachProfileListLen](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

## 8.1081.1 Detailed Description

This structure contains unpack get 3GPP configure item information.

## Parameters

	<i>LTEAttachProfile</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> </ul> </li> <li>• This setting is deprecated on MC/EM74xx</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
--	-------------------------	--

	<i>profileList</i>	<p>Profile List</p> <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 SLQSGet3GPPConfigItem() returns a default value 255 if no 3gpp configuration is present</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> <li>• Note: the 5th entry is currently ignored, please set it to zero</li> </ul>
out	<i>defaultPDN-Enabled</i>	<ul style="list-style-type: none"> <li>• 0 - disabled</li> <li>• 1 - enabled</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
out	<i>_3gppRelease</i>	<p>3GPP release</p> <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 toward)</li> <li>• 6 - Release_10 (In 9x30 and toward)</li> <li>• 7 - Release_11 (In 9x30 and toward)</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
	<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1081.2 Field Documentation

8.1081.2.1 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::\_3gppRelease

8.1081.2.2 uint8\_t unpack\_wds\_SLQSGet3GPPConfigItem\_t::defaultPDNEnabled

- 8.1081.2.3 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfile`
- 8.1081.2.4 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileList[24]`
- 8.1081.2.5 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::LTEAttachProfileListLen`
- 8.1081.2.6 `swi_uint256_t unpack_wds_SLQSGet3GPPConfigItem_t::ParamPresenceMask`
- 8.1081.2.7 `uint16_t unpack_wds_SLQSGet3GPPConfigItem_t::profileList[5]`

## 8.1082 `unpack_wds_SLQSGetCurrDataSystemStat_t` Struct Reference

### Data Fields

- `uint8_t` [prefNetwork](#)
- `uint8_t` [networkInfoLen](#)
- [currNetworkInfo](#) `currNetworkInfo` [255]
- [swi\\_uint256\\_t](#) `ParamPresenceMask`

### 8.1082.1 Detailed Description

Data System Status

#### Parameters

<i>prefNetwork</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>networkInfoLen</i>	<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> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>currNetworkInfo</i>	<ul style="list-style-type: none"> <li>• Network information</li> <li>• See <a href="#">currNetworkInfo</a> for more details</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1082.2 Field Documentation

- 8.1082.2.1 `currNetworkInfo` `unpack_wds_SLQSGetCurrDataSystemStat_t::currNetworkInfo[255]`
- 8.1082.2.2 `uint8_t` `unpack_wds_SLQSGetCurrDataSystemStat_t::networkInfoLen`

8.1082.2.3 swi\_uint256\_t unpack\_wds\_SLQSGetCurrDataSystemStat\_t::ParamPresenceMask

8.1082.2.4 uint8\_t unpack\_wds\_SLQSGetCurrDataSystemStat\_t::prefNetwork

## 8.1083 unpack\_wds\_SLQSGGetCurrentChannelRate\_t Struct Reference

### Data Fields

- uint32\_t [current\\_channel\\_tx\\_rate](#)
- uint32\_t [current\\_channel\\_rx\\_rate](#)
- uint32\_t [max\\_channel\\_tx\\_rate](#)
- uint32\_t [max\\_channel\\_rx\\_rate](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1083.1 Detailed Description

This structure contains unpack get current channel rate information.

#### Parameters

<i>current_channel_tx_rate</i>	<ul style="list-style-type: none"> <li>• Current Channel Tx Rate.</li> <li>• Instantaneous channel Tx rate in bits per second.</li> <li>• In 9x15, this is the total current channel rate for all PDNs combined.</li> <li>• In 9x30 and later, this is the channel rate for a specific PDN.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>current_channel_rx_rate</i>	<ul style="list-style-type: none"> <li>• Current Channel Rx Rate.</li> <li>• Instantaneous channel Rx rate in bits per second.</li> <li>• In 9x15, this is the total current channel rate for all PDNs combined.</li> <li>• In 9x30 and later, this is the channel rate for a specific PDN</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>max_channel_tx_rate</i>	<ul style="list-style-type: none"> <li>• Max Channel Tx Rate.</li> <li>• Maximum total Tx rate that modem is able to support in current serving system in bits per second.</li> <li>• In 9x15, this is a default hard coded value for the current serving system.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>max_channel_rx_rate</i>	<ul style="list-style-type: none"> <li>• Max Channel Rx Rate.</li> <li>• Maximum total Rx rate that modem is able to support in current serving system in bits per second.</li> <li>• In 9x15, this is a default hard coded value for the current serving system.</li> <li>• Bit to check in ParamPresenceMask - 1</li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1083.2 Field Documentation

8.1083.2.1 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_rx_rate`

8.1083.2.2 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::current_channel_tx_rate`

8.1083.2.3 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_rx_rate`

8.1083.2.4 `uint32_t unpack_wds_SLQSGGetCurrentChannelRate_t::max_channel_tx_rate`

8.1083.2.5 `swi_uint256_t unpack_wds_SLQSGGetCurrentChannelRate_t::ParamPresenceMask`

## 8.1084 unpack\_wds\_SLQSGGetDataBearerTechnology\_t Struct Reference

### Data Fields

- `uint8_t dataBearerMask`
- `qmiWSDDataBearerTechnology curDataBearerTechnology`
- `qmiWSDDataBearerTechnology lastCallDataBearerTechnology`
- `swi_uint256_t ParamPresenceMask`

### 8.1084.1 Detailed Description

Structure to hold the data bearer technology values

#### Parameters

<i>dataBearerMask[OUT]</i>	<ul style="list-style-type: none"> <li>• This bit mask indicates if data bearer information for the current and/or last call has been received from the device. If a bit is set, then the information is available in the corresponding structure i.e. the one provided by the caller. Refer to <a href="#">liteQmiDataBearerMasks</a> for bit-mask positions.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>curDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> <li>• current data bearer technology value.</li> <li>• See <a href="#">qmiWSDDataBearerTechnology</a></li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>lastCallDataBearerTechnology[OUT]</i>	<ul style="list-style-type: none"> <li>• last call data bearer technology value.</li> <li>• See <a href="#">qmiWSDDataBearerTechnology</a> <ul style="list-style-type: none"> <li>– NULL if the parameter is not required</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1084.2 Field Documentation

8.1084.2.1 `qmiWSDDataBearerTechnology unpack_wds_SLQSGGetDataBearerTechnology_t::curDataBearerTechnology`



8.1084.2.2 uint8\_t unpack\_wds\_SLQSGetDataBearerTechnology\_t::dataBearerMask

8.1084.2.3 qmiWSDDataBearerTechnology unpack\_wds\_SLQSGetDataBearerTechnology\_t::lastCallDataBearerTechnology

8.1084.2.4 swi\_uint256\_t unpack\_wds\_SLQSGetDataBearerTechnology\_t::ParamPresenceMask

## 8.1085 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](#)
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1085.1 Detailed Description

This structure contains the DUN Call Info response parameters

#### Parameters

<a href="#">connectionStatus</a>	<ul style="list-style-type: none"> <li>• See <a href="#">connectionStatus</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<a href="#">callEndReason</a>	<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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<a href="#">txOKBytesCount</a>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error</li> <li>• Returned only if a data call is up</li> <li>• Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<a href="#">rxOKBytesCount</a>	<ul style="list-style-type: none"> <li>• Number of bytes received without error</li> <li>• Returned only if a data call is up</li> <li>• Bit to check in ParamPresenceMask - <b>19</b></li> </ul>

<i>dormancyStatus</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> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>dataBearerTech</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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>channelRate</i>	<ul style="list-style-type: none"> <li>• See <a href="#">dunchannelRate</a> for more information</li> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>lastCallTXOK-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> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>lastCallRXOK-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> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>

<i>mdmCall-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> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>lastCallData-BearerTech</i>	<ul style="list-style-type: none"> <li>• Last Call Data Bearer Technology</li> <li>• Returned only if not in a call and when the previous call was made using DUN <ul style="list-style-type: none"> <li>– 0x01 - cdma2000 1X</li> <li>– 0x02 - cdma2000 HRPD (1xEV-DO)</li> <li>– 0x03 - GSM</li> <li>– 0x04 - UMTS</li> <li>– 0x05 - cdma200 HRPD (1xEV-DO Rev A)</li> <li>– 0x06 - EDGE</li> <li>– 0x07 - HSDPA and WCDMA</li> <li>– 0x08 - WCDMA and HSUPA</li> <li>– 0x09 - HSDPA and HSUPA</li> <li>– 0x0A - LTE</li> <li>– 0x0B - cdma2000 EHRPD</li> <li>– 0x0C - HSDPA+ and WCDMA</li> <li>– 0x0D - HSDPA+ and HSUPA</li> <li>– 0x0E - DC_HSDPA+ and WCDMA</li> <li>– 0x0F - DC_HSDPA+ and HSUPA</li> <li>– 0x10 - HSDPA+ and 64QAM</li> <li>– 0x11 - HSDPA+, 64QAM and HSUPA</li> <li>– 0x12 - TDSCDMA</li> <li>– 0x13 - TDSCDMA and HSDPA</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1085.2 Field Documentation

8.1085.2.1 `uint16_t unpack_wds_SLQSGetDUNCallInfo_t::callEndReason`

8.1085.2.2 `dunchannelRate unpack_wds_SLQSGetDUNCallInfo_t::channelRate`

8.1085.2.3 `connectionStatus unpack_wds_SLQSGetDUNCallInfo_t::connectionStatus`

8.1085.2.4 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dataBearerTech`

8.1085.2.5 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::dormancyStatus`

8.1085.2.6 `uint8_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallDataBearerTech`

8.1085.2.7 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallRXOKBytesCnt`

- 8.1085.2.8 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::lastCallTXOKBytesCnt`
- 8.1085.2.9 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::mdmCallDurationActive`
- 8.1085.2.10 `swi_uint256_t unpack_wds_SLQSGetDUNCallInfo_t::ParamPresenceMask`
- 8.1085.2.11 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::rxOKBytesCount`
- 8.1085.2.12 `uint64_t unpack_wds_SLQSGetDUNCallInfo_t::txOKBytesCount`

## 8.1086 `unpack_wds_SLQSGetProfileSettings_t` Struct Reference

### Data Fields

- [UnPackGetProfileSettingOut](#) \* [pProfileSettings](#)
- `uint8_t` [ProfileType](#)
- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

### 8.1086.1 Detailed Description

This structure contains the profile setting information of parameter `pOutput` for API `unpack_wds_SLQSGetProfileSettings`

#### Parameters

<i>pProfileSettings</i>	- Profile Settings
<i>ProfileType</i>	- Profile Type <ul style="list-style-type: none"> <li>• 0 - Profile 3GPP</li> <li>• 1 - Profile 3GPP2</li> </ul>
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1086.2 Field Documentation

- 8.1086.2.1 `swi_uint256_t unpack_wds_SLQSGetProfileSettings_t::ParamPresenceMask`
- 8.1086.2.2 `UnPackGetProfileSettingOut`\* `unpack_wds_SLQSGetProfileSettings_t::pProfileSettings`
- 8.1086.2.3 `uint8_t unpack_wds_SLQSGetProfileSettings_t::ProfileType`
- 8.1086.2.4 `uint16_t unpack_wds_SLQSGetProfileSettings_t::Tlvresult`

## 8.1087 `unpack_wds_SLQSGetProfileSettingsV2_t` Struct Reference

### Data Fields

- [UnPackGetProfileSettingOutV2](#) \* [pProfileSettings](#)
- `uint8_t` [ProfileType](#)
- `uint16_t` [Tlvresult](#)

- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1087.1 Detailed Description

This structure contains the profile setting information of parameter pOutput for API unpack\_wds\_SLQSGetProfileSettingsV2

#### Parameters

<i>pProfileSettings</i>	- Profile Settings
<i>ProfileType</i>	- Profile Type <ul style="list-style-type: none"> <li>• 0 - Profile 3GPP</li> <li>• 1 - Profile 3GPP2</li> </ul>
<i>Tlvresult</i>	- unpack Tlv Result.
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1087.2 Field Documentation

8.1087.2.1 [swi\\_uint256\\_t](#) unpack\_wds\_SLQSGetProfileSettingsV2\_t::ParamPresenceMask

8.1087.2.2 [UnPackGetProfileSettingOutV2\\*](#) unpack\_wds\_SLQSGetProfileSettingsV2\_t::pProfileSettings

8.1087.2.3 [uint8\\_t](#) unpack\_wds\_SLQSGetProfileSettingsV2\_t::ProfileType

8.1087.2.4 [uint16\\_t](#) unpack\_wds\_SLQSGetProfileSettingsV2\_t::Tlvresult

## 8.1088 unpack\_wds\_SLQSGetRuntimeSettings\_t Struct Reference

### Data Fields

- [uint32\\_t](#) [IPv4](#)
- [uint8\\_t](#) [ProfileName](#) [128]
- [uint32\\_t](#) [PDPTYPE](#)
- [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](#)
- [swi\\_uint256\\_t ParamPresenceMask](#)

### 8.1088.1 Detailed Description

This structure contains unpack get runtime settings information.

#### Parameters

<i>IPv4</i>	ipv4 address <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ProfileName</i>	profile name <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>PDPTtype</i>	PDP type <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>APNName</i>	APN name <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>PrimaryDNSV4</i>	primary dns IPV4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>SecondaryDNS-V4</i>	secondary dns IPV4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>UMTSGranted-QoS</i>	UMTS Granted QoS <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>GPRSGranted-QoS</i>	GPRS Granted QoS <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>Username</i>	username for authentication process <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>Authentication</i>	authentication for authentication process <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>ProfielID</i>	profile ID <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>31</b></li> </ul>
<i>GWAddressV4</i>	Gateway IPv4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>32</b></li> </ul>
<i>SubnetMaskV4</i>	Subnet mask IPV4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>33</b></li> </ul>
<i>PCSCFAddrPC-O</i>	PCSCF address PCO <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>34</b></li> </ul>

<i>ServerAddrList</i>	PCSCF server address list IPV4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>35</b></li> </ul>
<i>PCSCFFQDN-AddrList</i>	PCSCF FQDN address list IPV4 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>PrimaryDNSV6</i>	Primary DNS IPV6 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>39</b></li> </ul>
<i>SecondaryDNS-V6</i>	Secondary DNS IPV6 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>40</b></li> </ul>
<i>Mtu</i>	actual (runtime) Maximum Transfer Unit <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>41</b></li> </ul>
<i>DomainList</i>	domain list <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>
<i>IPFamily-Preference</i>	ip family preference, it could be IPV4 or IPV6 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>43</b></li> </ul>
<i>IMCNflag</i>	IM control flag, value: TRUE or FALSE <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>44</b></li> </ul>
<i>Technology</i>	technology on which current packet data session is in progress <ul style="list-style-type: none"> <li>• Values: <ul style="list-style-type: none"> <li>– 32767 - CDMA</li> <li>– 32764 - UMTS</li> <li>– 30592 - EPC</li> <li>– 30584 - modem link local</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>45</b></li> </ul>
<i>IPV6AddrInfo</i>	address information IPV6 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>37</b></li> </ul>
<i>IPV6GWAddr-Info</i>	gateway address information IPV6 <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>38</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1088.2 Field Documentation

8.1088.2.1 uint8\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::APNName[128]

8.1088.2.2 uint32\_t unpack\_wds\_SLQSGetRuntimeSettings\_t::Authentication

8.1088.2.3 struct wds\_DomainNameList unpack\_wds\_SLQSGetRuntimeSettings\_t::DomainList

8.1088.2.4 struct wds\_GPRSQoS unpack\_wds\_SLQSGetRuntimeSettings\_t::GPRSGrantedQoS

- 8.1088.2.5 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::GWAddressV4`
- 8.1088.2.6 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IMCNflag`
- 8.1088.2.7 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::IPFamilyPreference`
- 8.1088.2.8 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::IPv4`
- 8.1088.2.9 `struct wds_IPV6AddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6AddrInfo`
- 8.1088.2.10 `struct wds_IPV6GWAddressInfo unpack_wds_SLQSGetRuntimeSettings_t::IPv6GWAddrInfo`
- 8.1088.2.11 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::Mtu`
- 8.1088.2.12 `swi_uint256_t unpack_wds_SLQSGetRuntimeSettings_t::ParamPresenceMask`
- 8.1088.2.13 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::PCSCFAddrPCO`
- 8.1088.2.14 `struct wds_PCSCFFQDNAddressList unpack_wds_SLQSGetRuntimeSettings_t::PCSCFFQDNAddrList`
- 8.1088.2.15 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PDPTType`
- 8.1088.2.16 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV4`
- 8.1088.2.17 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::PrimaryDNSV6[8]`
- 8.1088.2.18 `struct wds_ProfileIdentifier unpack_wds_SLQSGetRuntimeSettings_t::ProfileID`
- 8.1088.2.19 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::ProfileName[128]`
- 8.1088.2.20 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV4`
- 8.1088.2.21 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::SecondaryDNSV6[8]`
- 8.1088.2.22 `struct wds_PCSCFIPv4ServerAddressList unpack_wds_SLQSGetRuntimeSettings_t::ServerAddrList`
- 8.1088.2.23 `uint32_t unpack_wds_SLQSGetRuntimeSettings_t::SubnetMaskV4`
- 8.1088.2.24 `uint16_t unpack_wds_SLQSGetRuntimeSettings_t::Technology`
- 8.1088.2.25 `LibPackUMTSQoS unpack_wds_SLQSGetRuntimeSettings_t::UMTSGrantedQoS`
- 8.1088.2.26 `uint8_t unpack_wds_SLQSGetRuntimeSettings_t::Username[128]`

## 8.1089 `unpack_wds_SLQSModifyProfile_t` Struct Reference

### Data Fields

- `uint16_t * pExtErrorCode`
- `swi_uint256_t ParamPresenceMask`

### 8.1089.1 Detailed Description

This structure contains out parameters for `unpack_wds_SLQSModifyProfile`



## 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> <li>Bit to check in ParamPresenceMask - <b>224</b></li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1089.2 Field Documentation

8.1089.2.1 `swi_uint256_t` unpack\_wds\_SLQSModifyProfile\_t::ParamPresenceMask8.1089.2.2 `uint16_t*` unpack\_wds\_SLQSModifyProfile\_t::pExtErrorCode

## 8.1090 unpack\_wds\_SLQSSetIPFamilyPreference\_t Struct Reference

## Data Fields

- `uint16_t` [Tlvresult](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.1090.1 Detailed Description

This structure contains unpack Set IP family preference information

## Parameters

<i>Tlvresult</i>	unpack result
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1090.2 Field Documentation

8.1090.2.1 `swi_uint256_t` unpack\_wds\_SLQSSetIPFamilyPreference\_t::ParamPresenceMask8.1090.2.2 `uint16_t` unpack\_wds\_SLQSSetIPFamilyPreference\_t::Tlvresult

## 8.1091 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](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1091.1 Detailed Description

This structure contains unpack set packet service status callback information.

#### Parameters

<i>conn_status</i>	connection status. <ul style="list-style-type: none"> <li>• Current link status. Values:               <ul style="list-style-type: none"> <li>– 1 - DISCONNECTED</li> <li>– 2 - CONNECTED</li> <li>– 3 - SUSPENDED</li> <li>– 4 - AUTHENTICATING</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>reconfigReqd</i>	Indicates whether the network interface on the host needs to be reconfigured. <ul style="list-style-type: none"> <li>• Values:               <ul style="list-style-type: none"> <li>– 0 - No need to reconfigure.</li> <li>– 1 - Reconfiguration required.</li> </ul> </li> </ul>

- Bit to check in ParamPresenceMask - **1**

#### Parameters

<i>sessionEnd-Reason</i>	Call End Reason <ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>verboseSessn-EndReasonType</i>	Verbose call end reason type <ul style="list-style-type: none"> <li>• Call end reason type. Values:               <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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>verboseSessn-EndReason</i>	Reason the call ended (verbose) <ul style="list-style-type: none"> <li>• See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>

<i>ipFamily</i>	<p>IP family of the packet data connection.</p> <ul style="list-style-type: none"> <li>Values <ul style="list-style-type: none"> <li>4 - IPv4</li> <li>6 - IPv6</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>techName</i>	<p>Technology name of the packet data connection.</p> <ul style="list-style-type: none"> <li>Values <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 EPC is a logical interface to support LTE/eHRPD handoff. It is returned if the device supports IP session continuity. Modem Link Local is an interface for transferring data between entities on the AP and modem.</li> </ul> </li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<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 <ul style="list-style-type: none"> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul> </li> </ul>
<i>ParamPresenceMask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1091.2 Field Documentation

8.1091.2.1 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::bearerID`

8.1091.2.2 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::conn_status`

8.1091.2.3 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ipFamily`

8.1091.2.4 `swi_uint256_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::ParamPresenceMask`

8.1091.2.5 `uint8_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::reconfigReqd`

8.1091.2.6 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::sessionEndReason`

8.1091.2.7 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::techName`

8.1091.2.8 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReason`

8.1091.2.9 `uint16_t unpack_wds_SLQSSetPacketSrvStatusCallback_t::verboseSessnEndReasonType`

## 8.1092 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]
- uint8\_t dBtechExtAvail
- uint32\_t dBtechnologyExt
- uint32\_t dBTechExtRatValue
- uint64\_t dBTechExtSoMask
- swi\_uint256\_t ParamPresenceMask

### 8.1092.1 Detailed Description

This structure contains unpack set WDS event callback information.

#### Parameters

<i>xferStatAvail</i>	transfer statistic available
<i>tx_bytes</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error</li> <li>• Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>rx_bytes</i>	<ul style="list-style-type: none"> <li>• Number of bytes received without error</li> <li>• Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>tx_pkts</i>	<ul style="list-style-type: none"> <li>• Number of packets transmitted without error</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>rx_pkts</i>	<ul style="list-style-type: none"> <li>• Number of packets received without error.</li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>mipstatAvail</i>	Mobile IP status available <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>mipStatus</i>	<ul style="list-style-type: none"> <li>• Status of the last MIP call (or attempt).</li> <li>• Values               <ul style="list-style-type: none"> <li>– 0x00 - Success</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>27</b></li> </ul>

<i>dBTechAvail</i>	Data Bearer technology available <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>dBTechnology</i>	<ul style="list-style-type: none"> <li>• Data Bearer technology               <ul style="list-style-type: none"> <li>– Values                   <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 - cdma2000 ® 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>* 0x14 - TDSCDMA and HSUPA</li> <li>* -1 - Unknown</li> </ul> </li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>dormancyStat-Avail</i>	Dormancy status available <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>dormancyStatus</i>	<ul style="list-style-type: none"> <li>• Dormancy status</li> <li>• Values               <ul style="list-style-type: none"> <li>– 1 - Traffic channel dormant</li> <li>– 2 - Traffic channel active</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>24</b></li> </ul>
<i>currDBTechAvail</i>	<ul style="list-style-type: none"> <li>• Current Data Bearer technology available</li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>

<i>ratMask</i>	<ul style="list-style-type: none"> <li>• RAT mask to indicate type of technology</li> <li>• Values <ul style="list-style-type: none"> <li>– 0x00 - DONT_CARE</li> <li>– 0x8000 - NULL_BEARER</li> </ul> </li> <li>• CDMA RAT mask <ul style="list-style-type: none"> <li>– 0x01 - CDMA_1X</li> <li>– 0x02 - EVDO_REV0</li> <li>– 0x05 - HRPD</li> <li>– 0x0B - EHRPD</li> </ul> </li> <li>• UMTS RAT mask <ul style="list-style-type: none"> <li>– 0x03 - GPRS</li> <li>– 0x04 - WCDMA</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>– 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> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>soMask</i>	<ul style="list-style-type: none"> <li>• <a href="#">SO Mask</a></li> <li>• Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>dataSysStatAvail</i>	<ul style="list-style-type: none"> <li>• Data System Status available</li> <li>• Bit to check in ParamPresenceMask - <b>36</b></li> </ul>
<i>prefNetwork</i>	<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>currNWInfo</i>	<ul style="list-style-type: none"> <li>• Current Network Info</li> <li>• see <a href="#">wds_currNetworkInfo</a> for more info</li> </ul>
<i>dBtechExtAvail</i>	<p>Data bearer technology extended available</p> <ul style="list-style-type: none"> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>

<i>dBtechnologyExt</i>	<ul style="list-style-type: none"> <li>• Data bearer technology Extended.</li> <li>• Values <ul style="list-style-type: none"> <li>– WDS_BEARER_TECH_NETWORK_3GPP (0) - 3GPP</li> <li>– WDS_BEARER_TECH_NETWORK_3GPP2 (1) - 3GPP2</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>
<i>dBTechExtRat-Value</i>	<ul style="list-style-type: none"> <li>• Data bearer technology Extended RAT Value</li> <li>• Values <ul style="list-style-type: none"> <li>– WDS_BEARER_TECH_RAT_EX_NULL_BEARER (0x00) - NULL bearer</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_WCDMA (0x01) - 3GPP WCDMA</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_GERAN (0x02) - 3GPP GERAN</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_LTE (0x03) - 3GPP LTE</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_TDSCDMA (0x04) - 3GPP TDSCDMA</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_WLAN (0x05) - 3GPP WLAN</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP_MAX (0x64) - 3GPP maximum</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP2_1X (0x65) - 3GPP2 1X</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP2_HRPD (0x66) - 3GPP2 HRPD</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP2_EHRPD (0x67) - 3GPP2 EHRPD</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP2_WLAN (0x68) - 3GPP2 WLAN</li> <li>– WDS_BEARER_TECH_RAT_EX_3GPP2_MAX (0xC8) - 3GPP2 maximum</li> </ul> </li> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>

<i>dBTechExtSo-Mask</i>	<ul style="list-style-type: none"> <li>• Data bearer technology Extended SO Mask</li> <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> <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> </li> <li>• Bit to check in ParamPresenceMask - <b>42</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1092.2 Field Documentation

8.1092.2.1 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::currDBTechAvail`

8.1092.2.2 `wds_currNetworkInfo unpack_wds_SLQSSetWdsEventCallback_ind_t::currNWInfo[255]`

8.1092.2.3 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dataSysStatAvail`

8.1092.2.4 `uint8_t unpack_wds_SLQSSetWdsEventCallback_ind_t::dBTechAvail`



- 8.1092.2.5 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dBtechExtAvail
- 8.1092.2.6 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dBTechExtRatValue
- 8.1092.2.7 uint64\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dBTechExtSoMask
- 8.1092.2.8 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::DBTechnology
- 8.1092.2.9 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dBtechnologyExt
- 8.1092.2.10 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dormancyStatAvail
- 8.1092.2.11 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::dormancyStatus
- 8.1092.2.12 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::mipstatAvail
- 8.1092.2.13 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::mipStatus
- 8.1092.2.14 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::netInfoLen
- 8.1092.2.15 swi\_uint256\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::ParamPresenceMask
- 8.1092.2.16 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::prefNetwork
- 8.1092.2.17 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::ratMask
- 8.1092.2.18 uint64\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::rx\_bytes
- 8.1092.2.19 uint64\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::rx\_pkts
- 8.1092.2.20 uint32\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::soMask
- 8.1092.2.21 uint64\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::tx\_bytes
- 8.1092.2.22 uint64\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::tx\_pkts
- 8.1092.2.23 uint8\_t unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t::xferStatAvail

## 8.1093 unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t Struct Reference

### Data Fields

- [wdsDhcpv4HwConfig](#) \* [pHwConfig](#)
- [wdsDhcpv4OptionList](#) \* [pRequestOptionList](#)
- [swi\\_uint256\\_t](#) [ParamPresenceMask](#)

### 8.1093.1 Detailed Description

WDS SWI DHCPv4 Config Structure

## Parameters

<i>pHwConfig</i>	<ul style="list-style-type: none"> <li>• pointer to HW Config structure</li> <li>• See <a href="#">wdsDhcpv4HwConfig</a></li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>pRequestOption-List</i>	<ul style="list-style-type: none"> <li>• pointer to Option List structure to be sent in DHCP request</li> <li>• See <a href="#">wdsDhcpv4OptionList</a></li> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1093.2 Field Documentation

8.1093.2.1 `swi_uint256_t unpack_wds_SLQSSGetDHCPv4ClientConfig_t::ParamPresenceMask`

8.1093.2.2 `wdsDhcpv4HwConfig* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pHwConfig`

8.1093.2.3 `wdsDhcpv4OptionList* unpack_wds_SLQSSGetDHCPv4ClientConfig_t::pRequestOptionList`

8.1094 `unpack_wds_SLQSSGetLoopback_t` Struct Reference

## Data Fields

- `uint8_t` [ByteLoopbackMode](#)
- `uint8_t` [ByteLoopbackMultiplier](#)
- `swi_uint256_t` [ParamPresenceMask](#)

## 8.1094.1 Detailed Description

This structure contains unpack Get loopback 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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1094.2 Field Documentation

8.1094.2.1 `uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMode`

8.1094.2.2 `uint8_t unpack_wds_SLQSSGetLoopback_t::ByteLoopbackMultiplier`

8.1094.2.3 `swi_uint256_t unpack_wds_SLQSSGetLoopback_t::ParamPresenceMask`

## 8.1095 unpack\_wds\_SLQSSStartDataSession\_t Struct Reference

### Data Fields

- `uint32_t * psid`
- `uint32_t * pFailureReason`
- `uint32_t * pVerboseFailReasonType`
- `uint32_t * pVerboseFailureReason`
- `swi_uint256_t ParamPresenceMask`

### 8.1095.1 Detailed Description

This structure contains unpack Start Data Session Information.

#### Parameters

<i>psid</i>	<ul style="list-style-type: none"> <li>• Assigned session ID when starting a data session</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></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> <li>• Bit to check in ParamPresenceMask - <b>16</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></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> <li>• Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

## 8.1095.2 Field Documentation

8.1095.2.1 `swi_uint256_t unpack_wds_SLQSSStartDataSession_t::ParamPresenceMask`

8.1095.2.2 `uint32_t* unpack_wds_SLQSSStartDataSession_t::pFailureReason`

8.1095.2.3 `uint32_t*` `unpack_wds_SLQSSstartDataSession_t::psid`

8.1095.2.4 `uint32_t*` `unpack_wds_SLQSSstartDataSession_t::pVerboseFailReasonType`

8.1095.2.5 `uint32_t*` `unpack_wds_SLQSSstartDataSession_t::pVerboseFailureReason`

## 8.1096 `unpack_wds_SLQSSwiProfileChangeCallback_Ind_t` Struct Reference

### Data Fields

- [wds\\_profileChange](#) ProfileTlv
- [wds\\_sourceOfChange](#) srcTlv
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1096.1 Detailed Description

Structure for SwiProfileChange unpack.

#### Parameters

<i>ProfileTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">wds_profileChange</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>1</b></li> </ul>
<i>srcTlv</i>	<ul style="list-style-type: none"> <li>• See <a href="#">wds_sourceOfChange</a> for more information.</li> <li>• Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>• bitmask representation to indicate valid parameters.</li> </ul>

### 8.1096.2 Field Documentation

8.1096.2.1 `swi_uint256_t` `unpack_wds_SLQSSwiProfileChangeCallback_Ind_t::ParamPresenceMask`

8.1096.2.2 `wds_profileChange` `unpack_wds_SLQSSwiProfileChangeCallback_Ind_t::ProfileTlv`

8.1096.2.3 `wds_sourceOfChange` `unpack_wds_SLQSSwiProfileChangeCallback_Ind_t::srcTlv`

## 8.1097 `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`
- `ipv6AddressInfo` `ipv6Address`
- `ipv6AddressInfo` `ipv6GWAddress`

- uint16\_t [prDNSIPv6Address](#) [8]
- uint16\_t [seDNSIPv6Address](#) [8]
- uint32\_t [prPCSCFIPv4Address](#)
- uint32\_t [sePCSCFIPv4Address](#)
- uint16\_t [prPCSCFIPv6Address](#) [8]
- uint16\_t [sePCSCFIPv6Address](#) [8]
- [swi\\_uint256\\_t](#) ParamPresenceMask

### 8.1097.1 Detailed Description

This structure contains the unpack parameters retrieved by the API `unpack_wds_SLQSWdsSwiPDPRuntimeSettings`

#### Parameters

<i>contextId</i>	(optional) <ul style="list-style-type: none"> <li>Context Identifier</li> <li>Bit to check in ParamPresenceMask - <b>16</b></li> </ul>
<i>bearerId</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.</li> <li>Bit to check in ParamPresenceMask - <b>17</b></li> </ul>
<i>apnName</i>	(optional) <ul style="list-style-type: none"> <li>APN name associated with the context id</li> <li>Bit to check in ParamPresenceMask - <b>18</b></li> </ul>
<i>ipv4Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv4 Address</li> <li>Bit to check in ParamPresenceMask - <b>19</b></li> </ul>
<i>ipv4GWAddress</i>	(optional) <ul style="list-style-type: none"> <li>IPv4 Gateway Address</li> <li>Bit to check in ParamPresenceMask - <b>20</b></li> </ul>
<i>prDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary DNS IPv4 Address</li> <li>Bit to check in ParamPresenceMask - <b>21</b></li> </ul>
<i>seDNSIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary DNS IPv4 Address</li> <li>Bit to check in ParamPresenceMask - <b>22</b></li> </ul>
<i>ipv6Address</i>	(optional) <ul style="list-style-type: none"> <li>IPv6 Address</li> <li>Bit to check in ParamPresenceMask - <b>23</b></li> </ul>
<i>ipv6GWAddress</i>	(optional) <ul style="list-style-type: none"> <li>IPv6 Gateway Address</li> <li>See <a href="#">ipv6AddressInfo</a> for more information</li> <li>Bit to check in ParamPresenceMask - <b>24</b></li> </ul>

<i>prDNSIPv6-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> <li>Bit to check in ParamPresenceMask - <b>25</b></li> </ul>
<i>seDNSIPv6-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> <li>Bit to check in ParamPresenceMask - <b>26</b></li> </ul>
<i>prPCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Primary PCSCF IPv4 Address</li> <li>Bit to check in ParamPresenceMask - <b>27</b></li> </ul>
<i>sePCSCFIPv4-Address</i>	(optional) <ul style="list-style-type: none"> <li>Secondary PCSCF IPv4 Address</li> <li>Bit to check in ParamPresenceMask - <b>28</b></li> </ul>
<i>prPCSCFIPv6-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> <li>Bit to check in ParamPresenceMask - <b>29</b></li> </ul>
<i>sePCSCFIPv6-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> <li>Bit to check in ParamPresenceMask - <b>30</b></li> </ul>
<i>ParamPresence-Mask</i>	<ul style="list-style-type: none"> <li>bitmask representation to indicate valid parameters.</li> </ul>

## 8.1097.2 Field Documentation

8.1097.2.1 `int8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::apnName[100]`

8.1097.2.2 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::bearerId`

8.1097.2.3 `uint8_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::contextId`

8.1097.2.4 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4Address`

8.1097.2.5 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv4GWAddress`

8.1097.2.6 `ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6Address`

8.1097.2.7 `ipv6AddressInfo unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ipv6GWAddress`

8.1097.2.8 `swi_uint256_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::ParamPresenceMask`

8.1097.2.9 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv4Address`

- 8.1097.2.10 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prDNSIPv6Address[8]`
- 8.1097.2.11 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv4Address`
- 8.1097.2.12 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::prPCSCFIPv6Address[8]`
- 8.1097.2.13 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv4Address`
- 8.1097.2.14 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::seDNSIPv6Address[8]`
- 8.1097.2.15 `uint32_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv4Address`
- 8.1097.2.16 `uint16_t unpack_wds_SLQSWdsSwiPDPRuntimeSettings_t::sePCSCFIPv6Address[8]`

## 8.1098 UnPackGetProfileSettingOut Struct Reference

### Data Fields

- [UnpackQmiProfileInfo curProfile](#)
- `uint16_t *` [pExtErrCode](#)

### 8.1098.1 Detailed Description

This structure contains the profile settings output

#### Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> <li>• Structure containing details of the current profile</li> <li>• See <a href="#">UnpackQmiProfileInfo</a></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> <li>• Bit to check in ParamPresenceMask - <b>224</b></li> </ul>

### 8.1098.2 Field Documentation

8.1098.2.1 [UnpackQmiProfileInfo](#) `UnPackGetProfileSettingOut::curProfile`

8.1098.2.2 `uint16_t*` `UnPackGetProfileSettingOut::pExtErrCode`

## 8.1099 UnPackGetProfileSettingOutV2 Struct Reference

### Data Fields

- [UnpackQmiProfileInfoV2 curProfile](#)
- `uint16_t *` [pExtErrCode](#)

### 8.1099.1 Detailed Description

This structure contains the profile settings output

#### Parameters

<i>curProfile</i>	<ul style="list-style-type: none"> <li>Structure containing details of the current profile</li> <li>See <a href="#">UnpackQmiProfileInfoV2</a></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> <li>Bit to check in ParamPresenceMask - <b>224</b></li> </ul>

### 8.1099.2 Field Documentation

8.1099.2.1 [UnpackQmiProfileInfoV2](#) [UnPackGetProfileSettingOutV2::curProfile](#)

8.1099.2.2 [uint16\\_t\\*](#) [UnPackGetProfileSettingOutV2::pExtErrCode](#)

## 8.1100 UnpackSwiAvmsEventReportBinaryUpdateSessionInfo Struct Reference

### Data Fields

- [uint8\\_t](#) [bBinaryType](#)
- [uint8\\_t](#) [bState](#)
- [uint8\\_t](#) [bUserInputRequest](#)
- [uint16\\_t](#) [wUserInputTimeout](#)
- [uint32\\_t](#) [ulPkgDownloadSize](#)
- [uint32\\_t](#) [ulPkgDownloadComplete](#)
- [uint16\\_t](#) [wUpdateCompeteStatus](#)
- [uint8\\_t](#) [bSernity](#)
- [uint16\\_t](#) [wVersionLength](#)
- [uint8\\_t](#) [szVersion](#) [128]
- [uint16\\_t](#) [wNameLength](#)
- [uint8\\_t](#) [szName](#) [128]
- [uint16\\_t](#) [wDescriptionLength](#)
- [uint8\\_t](#) [szDescription](#) [1024]
- [uint8\\_t](#) [TlvPresent](#)

### 8.1100.1 Detailed Description

This structure contains unpack event report binary update session information parameters.



## Parameters

<i>bBinaryType</i>	: Type <ul style="list-style-type: none"> <li>• 1 - Firmware.</li> <li>• 2 - User App.</li> <li>• 3 - Legato Framework.</li> </ul>
<i>bStat</i>	: State <ul style="list-style-type: none"> <li>• 0x01 - No binary update available.</li> <li>• 0x02 - Query binary Download.</li> <li>• 0x03 - Binary Downloading.</li> <li>• 0x04 - Binary downloaded.</li> <li>• 0x05 -Query Binary Update.</li> <li>• 0x06 - Binary updating.</li> <li>• 0x07 - Binary updated.</li> </ul>
<i>bUserInput-Request</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>wUserInput-Timeout</i>	Timeout for user input in minutes. A value of 0 means no time-out.
<i>ulPkgDownload-Size</i>	The size (in bytes) of the update package
<i>ulPkgDownload-Complete</i>	The number of bytes being downloaded. For downloading state, this value shall be > 0 and incremented toward the pkg_dload_size. For other states, the value shall be 0 as it is meaningless.
<i>wUpdate-CompeteStatus</i>	Result code.This field should be looked at only when the AVMS session is complete.
<i>bSerty</i>	Serty. <ul style="list-style-type: none"> <li>• 0x01 - Mandatory.</li> <li>• 0x02 - Optional.</li> </ul>
<i>wVersionLength</i>	Length of FW Version string in bytes.
<i>szVersion</i>	FW Version string in ASCII.
<i>wNameLength</i>	Length Package Name string in bytes.
<i>szName</i>	Package Name in UCS2.
<i>wDescription-Length</i>	Length of description in bytes.
<i>szDescription</i>	Description of Update Package in USC2.
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

## 8.1100.2 Field Documentation

8.1100.2.1 uint8\_t UnpackSviAvmsEventReportBinaryUpdateSessionInfo::bBinaryType

8.1100.2.2 uint8\_t UnpackSviAvmsEventReportBinaryUpdateSessionInfo::bSerty

8.1100.2.3 uint8\_t UnpackSviAvmsEventReportBinaryUpdateSessionInfo::bState

8.1100.2.4 uint8\_t UnpackSviAvmsEventReportBinaryUpdateSessionInfo::bUserInputRequest

8.1100.2.5 uint8\_t UnpackSviAvmsEventReportBinaryUpdateSessionInfo::szDescription[1024]

- 8.1100.2.6 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szName[128]`
- 8.1100.2.7 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::szVersion[128]`
- 8.1100.2.8 `uint8_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::TlvPresent`
- 8.1100.2.9 `uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadComplete`
- 8.1100.2.10 `uint32_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::ulPkgDownloadSize`
- 8.1100.2.11 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wDescriptionLength`
- 8.1100.2.12 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wNameLength`
- 8.1100.2.13 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUpdateCompeteStatus`
- 8.1100.2.14 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wUserInputTimeout`
- 8.1100.2.15 `uint16_t UnpackSwiAvmsEventReportBinaryUpdateSessionInfo::wVersionLength`

## 8.1101 UnpackSwiAvmsEventReportConfig Struct Reference

### Data Fields

- `uint8_t bState`
- `uint8_t bUserInputRequest`
- `uint16_t wUserInputTimeout`
- `uint16_t wAlertMsgLength`
- `uint8_t szAlertMsg [200]`
- `uint8_t TlvPresent`

### 8.1101.1 Detailed Description

This structure contains unpack event report configure parameters.

#### Parameters

<i>bState</i>	State. <ul style="list-style-type: none"> <li>• 0x01 - AVMS Read Request.</li> <li>• 0x02 - AVMS Change Request.</li> <li>• 0x03 - AVMS Config Complete.</li> </ul>
<i>bUserInput-Request</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>wUserInput-Timeout</i>	Timeout for user input in seconds. A value of 0 means no time-out
<i>wAlertMsg-Length</i>	Length of Alert message string in bytes.
<i>szAlertMsg</i>	Alert message in UCS2.
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

### 8.1101.2 Field Documentation

- 8.1101.2.1 `uint8_t` `UnpackSwiAvmsEventReportConfig::bState`
- 8.1101.2.2 `uint8_t` `UnpackSwiAvmsEventReportConfig::bUserInputRequest`
- 8.1101.2.3 `uint8_t` `UnpackSwiAvmsEventReportConfig::szAlertMsg[200]`
- 8.1101.2.4 `uint8_t` `UnpackSwiAvmsEventReportConfig::TlvPresent`
- 8.1101.2.5 `uint16_t` `UnpackSwiAvmsEventReportConfig::wAlertMsgLength`
- 8.1101.2.6 `uint16_t` `UnpackSwiAvmsEventReportConfig::wUserInputTimeout`

## 8.1102 UnpackSwiAvmsEventReportConnectionRequest Struct Reference

### Data Fields

- `uint8_t` `bUserInputRequest`
- `uint16_t` `wUserInputTimeout`
- `uint8_t` `TlvPresent`

### 8.1102.1 Detailed Description

This structure contains unpack event report connection request parameters.

#### Parameters

<i>bUserInput-Request</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>bUserInput-Request</i>	Timeout for user input in minutes. A value of 0 means no time-out.
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

### 8.1102.2 Field Documentation

- 8.1102.2.1 `uint8_t` `UnpackSwiAvmsEventReportConnectionRequest::bUserInputRequest`
- 8.1102.2.2 `uint8_t` `UnpackSwiAvmsEventReportConnectionRequest::TlvPresent`
- 8.1102.2.3 `uint16_t` `UnpackSwiAvmsEventReportConnectionRequest::wUserInputTimeout`

## 8.1103 UnpackSwiAvmsEventReportDataSessionStatus Struct Reference

### Data Fields

- `uint8_t` `bType`
- `uint16_t` `wErrorCode`
- `uint8_t` `TlvPresent`

### 8.1103.1 Detailed Description

This structure contains unpack evnet report data session status parameters.

#### Parameters

<i>bType</i>	Notification type. <ul style="list-style-type: none"><li>• 0: Data session closed.</li><li>• 1: Data session activated.</li><li>• 2: Register made. -3: Data session error.</li></ul>
<i>wErrorCode</i>	LWM2M Session error code. <ul style="list-style-type: none"><li>• 0x0000: none</li></ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

### 8.1103.2 Field Documentation

8.1103.2.1 `uint8_t UnpackSwiAvmsEventReportDataSessionStatus::bType`

8.1103.2.2 `uint8_t UnpackSwiAvmsEventReportDataSessionStatus::TlvPresent`

8.1103.2.3 `uint16_t UnpackSwiAvmsEventReportDataSessionStatus::wErrorCode`

## 8.1104 UnpackSwiAvmsEventReportHTTPStatus Struct Reference

#### Data Fields

- `uint16_t wHTTPStatus`
- `uint8_t TlvPresent`

### 8.1104.1 Detailed Description

This structure contains unpack event report HTTP status parameters.

## Parameters

<i>wHTTPStatus</i>	See RFC 7231. <ul style="list-style-type: none"> <li>• 100 - Continue</li> <li>• 101 - Switching Protocols</li> <li>• 200 - OK</li> <li>• 201 - Created</li> <li>• 202 - Accepted</li> <li>• 203 - Non-Authoritative Information</li> <li>• 204 - No Content</li> <li>• 205 - Reset Content</li> <li>• 206 - Partial Content</li> <li>• 300 - Multiple Choices</li> <li>• 301 - Moved Permanently</li> <li>• 302 - Found</li> <li>• 303 - See Other</li> <li>• 304 - Not Modified</li> <li>• 305 - Use Proxy</li> <li>• 307 - Temporary Redirect</li> <li>• 400 - Bad Request</li> <li>• 401 - Unauthorized</li> <li>• 402 - Payment Required</li> <li>• 403 - Forbidden</li> <li>• 404 - Not Found</li> <li>• 405 - Method Not Allowed</li> <li>• 406 - Not Acceptable</li> <li>• 407 - Proxy Authentication Required</li> <li>• 408 - Request Timeout</li> <li>• 409 - Conflict</li> <li>• 410 - Gone</li> <li>• 411 - Length Required</li> <li>• 412 - Precondition Failed</li> <li>• 413 - Payload Too Large</li> <li>• 414 - URI Too Long</li> <li>• 415 - Unsupported Media Type</li> <li>• 416 - Range Not Satisfiable</li> <li>• 417 - Expectation Failed</li> <li>• 426 - Upgrade Required</li> <li>• 500 - Internal Server Error</li> <li>• 501 - Not Implemented</li> <li>• 502 - Bad Gateway</li> <li>• 503 - Service Unavailable</li> <li>• 504 - Gateway Timeout</li> <li>• 505 - HTTP Version Not Supported</li> </ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

## 8.1104.2 Field Documentation

## 8.1104.2.1 uint8\_t UnpackSwiAvmsEventReportHTTPStatus::TlvPresent

8.1104.2.2 uint16\_t UnpackSwiAvmsEventReportHTTPStatus::wHTTPStatus

## 8.1105 UnpackSwiAvmsEventReportNotification Struct Reference

### Data Fields

- uint8\_t [bNotification](#)
- uint16\_t [wSessionStatus](#)
- uint8\_t [TlvPresent](#)

### 8.1105.1 Detailed Description

This structure contains unpack event report notification parameters.

#### Parameters

<i>bNotification</i>	Notification. <ul style="list-style-type: none"> <li>• 0x14 - Module starts sending data to server.</li> <li>• 0x15 - Authentication with the server.</li> <li>• 0x16 - session with the server is ended.</li> </ul>
<i>wSessionStatus</i>	This field will set to the session status for notifications that occur at the end of a session, zero for all other notifications. <ul style="list-style-type: none"> <li>• 0x0000: Successful: Session succeeded</li> <li>• 0x0001: Break: Session succeeded</li> <li>• 0x0002: Large Object Handled: Session succeeded</li> <li>• 0x0003: No status: Session succeeded</li> <li>• 0x0004: No more commands: Session succeeded</li> <li>• 0x0005: User cancel: Session cancelled</li> <li>• 0x0100-0x01FF: General errors</li> <li>• 0x0200-0x02FF: Syncml errors</li> <li>• 0x0300-0x03FF: Authentication errors</li> <li>• 0x0400-0x04FF: Protocol errors</li> <li>• 0x0500-0x05FF: Tree errors (DM Only)</li> <li>• 0x0600-0x06FF: Not applicable</li> <li>• 0x0700-0x07FF: Trigger errors</li> <li>• 0x0800-0x08FF: FUMO errors</li> <li>• 0x0900-0x09FF: Communication errors</li> <li>• 0x0A00-0x0AFF: Parsing errors</li> <li>• 0x0B00-0x0CFF: Not applicable</li> <li>• 0x7F00-0x7F12: Insignia errors</li> <li>• 0x7F13: Illegal text: Text received contains illegal characters</li> <li>• 0x7F14: Download failure: Failed to download FOTA image</li> <li>• 0x7F15: Empty session: Session ran successfully, but no information was updated</li> <li>• 0x7F16: Factory reset successful: Factory reset succeeded</li> <li>• 0x7F17: Factory reset fail: Factory reset failed</li> </ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

### 8.1105.2 Field Documentation

8.1105.2.1 uint8\_t UnpackSwiAvmsEventReportNotification::bNotification

8.1105.2.2 uint8\_t UnpackSwiAvmsEventReportNotification::TlvPresent

8.1105.2.3 uint16\_t UnpackSwiAvmsEventReportNotification::wSessionStatus

## 8.1106 UnpackSwiAvmsEventReportPackageID Struct Reference

### Data Fields

- uint8\_t [bPackageID](#)
- uint8\_t [TlvPresent](#)

#### 8.1106.1 Detailed Description

This structure contains unpack event report package ID parameters.

##### Parameters

<i>bPackageID</i>	Package ID of the application binary that this AVMS_EVENT_ID notification is for.
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

#### 8.1106.2 Field Documentation

8.1106.2.1 uint8\_t UnpackSwiAvmsEventReportPackageID::bPackageID

8.1106.2.2 uint8\_t UnpackSwiAvmsEventReportPackageID::TlvPresent

## 8.1107 UnpackSwiAvmsEventReportRegStatus Struct Reference

### Data Fields

- uint8\_t [bRegStatus](#)
- uint8\_t [TlvPresent](#)

#### 8.1107.1 Detailed Description

This structure contains unpack event report registration status parameters.

##### Parameters

<i>bRegStatus</i>	LWM2M Registration status. <ul style="list-style-type: none"><li>• 0: Need Bootstrap.</li><li>• 1: Bootstrap made.</li><li>• 2: Register made. -3: Update made.</li></ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

#### 8.1107.2 Field Documentation

8.1107.2.1 uint8\_t UnpackSwiAvmsEventReportRegStatus::bRegStatus

8.1107.2.2 `uint8_t UnpackSwiAvmsEventReportRegStatus::TlvPresent`

## 8.1108 UnpackSwiAvmsEventReportSessionType Struct Reference

### Data Fields

- `uint8_t bType`
- `uint8_t TlvPresent`

### 8.1108.1 Detailed Description

This structure contains unpack event report session type parameters.

#### Parameters

<i>bType</i>	Session Type. <ul style="list-style-type: none"><li>• 0: Bootstrap session.</li><li>• 1. DM session</li></ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response

### 8.1108.2 Field Documentation

8.1108.2.1 `uint8_t UnpackSwiAvmsEventReportSessionType::bType`

8.1108.2.2 `uint8_t UnpackSwiAvmsEventReportSessionType::TlvPresent`

## 8.1109 UnpackSwiAvmsEventReportWAMSParamChange Struct Reference

### Data Fields

- `uint16_t wWamsChangeMask`
- `uint8_t TlvPresent`

### 8.1109.1 Detailed Description

This structure contains unpack event report WAMS parameter change parameters.

#### Parameters

<i>wWamsChangeMask</i>	Mask of WAMS parameters changed. <ul style="list-style-type: none"><li>• By default set to 0xFF for all changes.</li><li>• 0x01 - device_login</li><li>• 0x02 - device_MD5_key</li><li>• 0x04 - server_login</li><li>• 0x08 - server_MD5_key</li><li>• 0x10 - server_URL</li><li>• 0x20 - Nonce</li><li>• 0x40 - Application key</li></ul>
<i>TlvPresent</i>	Boolean indicating the presence of the TLV in the QMI response



## 8.1109.2 Field Documentation

8.1109.2.1 `uint8_t` `UnpackSwiAvmsEventReportWAMSParamChange::TlvPresent`

8.1109.2.2 `uint16_t` `UnpackSwiAvmsEventReportWAMSParamChange::wWamsChangeMask`

## 8.1110 unpackWdsProfileParam Union Reference

### Data Fields

- [LibpackProfile3GPP](#) `SlqsProfile3GPP`
- [LibpackProfile3GPP2](#) `SlqsProfile3GPP2`

### 8.1110.1 Detailed Description

This union `WdsProfileParam` consists of `Profile3GPP` and `Profile3GPP2` out of which one will be used to create profile.

#### Parameters

<i>SlqsProfile3GPP</i>	3GPP profile <ul style="list-style-type: none"> <li>• See <a href="#">LibpackProfile3GPP</a></li> </ul>
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> <li>• See <a href="#">LibpackProfile3GPP2</a></li> </ul>

## 8.1110.2 Field Documentation

8.1110.2.1 `LibpackProfile3GPP` `unpackWdsProfileParam::SlqsProfile3GPP`

8.1110.2.2 `LibpackProfile3GPP2` `unpackWdsProfileParam::SlqsProfile3GPP2`

## 8.1111 unpackWdsProfileParamV2 Union Reference

### Data Fields

- [LibpackProfile3GPPV2](#) `SlqsProfile3GPP`
- [LibpackProfile3GPP2](#) `SlqsProfile3GPP2`

### 8.1111.1 Detailed Description

This union `WdsProfileParam` consists of `Profile3GPP` and `Profile3GPP2` out of which one will be used to create profile.

#### Parameters

<i>SlqsProfile3GPP</i>	3GPP profile <ul style="list-style-type: none"> <li>• See <a href="#">LibpackProfile3GPP</a></li> </ul>
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> <li>• See <a href="#">LibpackProfile3GPP2</a></li> </ul>

## 8.1111.2 Field Documentation

8.1111.2.1 `LibpackProfile3GPPV2 unpackWdsProfileParamV2::SlqsProfile3GPP`

8.1111.2.2 `LibpackProfile3GPP2 unpackWdsProfileParamV2::SlqsProfile3GPP2`

## 8.1112 voice\_airTimer Struct Reference

### Data Fields

- `uint8_t` [namID](#)
- `uint32_t` [airTimerValue](#)

### 8.1112.1 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>

## 8.1112.2 Field Documentation

8.1112.2.1 `uint32_t` `voice_airTimer::airTimerValue`

8.1112.2.2 `uint8_t` `voice_airTimer::namID`

## 8.1113 voice\_allCallsAlphaIDInfo Struct Reference

### Data Fields

- `uint8_t` [callID](#)
- [voice\\_alphaIDInfo AlphaIDInfo](#)

### 8.1113.1 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>AlphaDInfo</i>	<ul style="list-style-type: none"><li>• See <a href="#">voice_alphaDInfo</a> for more information.</li></ul>
-------------------	--

## 8.1113.2 Field Documentation

8.1113.2.1 `voice_alphaDInfo voice_allCallsAlphaDInfo::AlphaDInfo`

8.1113.2.2 `uint8_t voice_allCallsAlphaDInfo::callID`

## 8.1114 voice\_allCallsDiagInfo Struct Reference

### Data Fields

- `uint8_t callID`
- [voice\\_diagInfo DiagInfo](#)

### 8.1114.1 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="#">voice_diagInfo</a> for more information.</li></ul>

## 8.1114.2 Field Documentation

8.1114.2.1 `uint8_t voice_allCallsDiagInfo::callID`

8.1114.2.2 `voice_diagInfo voice_allCallsDiagInfo::DiagInfo`

## 8.1115 voice\_allCallsUUSInfo Struct Reference

### Data Fields

- `uint8_t callID`
- [voice\\_UUSInfo uusInfo](#)

### 8.1115.1 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="#">voice_UUSInfo</a> for more information.</li> </ul>
----------------	---

## 8.1115.2 Field Documentation

8.1115.2.1 `uint8_t voice_allCallsUUSInfo::callID`

8.1115.2.2 `voice_UUSInfo voice_allCallsUUSInfo::uusInfo`

## 8.1116 voice\_alphalDInfo Struct Reference

### Data Fields

- `uint8_t alphaDcs`
- `uint8_t alphaLen`
- `uint8_t alphaText [255]`

### 8.1116.1 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[<small>MAX_VOICE_DESCRIPTION_LENGTH</small>]</i>	<ul style="list-style-type: none"> <li>• Data encoded as per the alpha_dcs</li> </ul>

## 8.1116.2 Field Documentation

8.1116.2.1 `uint8_t voice_alphalDInfo::alphaDcs`

8.1116.2.2 `uint8_t voice_alphalDInfo::alphaLen`

8.1116.2.3 `uint8_t voice_alphalDInfo::alphaText[255]`

## 8.1117 voice\_arrAlertingPattern Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- uint8\_t [callID](#) [20]
- uint32\_t [alertingPattern](#) [20]

## 8.1117.1 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> [VOICE_ - MAX_NO_OF_ - CALLS]	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>alertingPattern</i> [-VOICE_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>

## 8.1117.2 Field Documentation

8.1117.2.1 uint32\_t voice\_arrAlertingPattern::alertingPattern[20]

8.1117.2.2 uint8\_t voice\_arrAlertingPattern::callID[20]

8.1117.2.3 uint8\_t voice\_arrAlertingPattern::numInstances

## 8.1118 voice\_arrAlertingType Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- uint8\_t [callID](#) [20]
- uint8\_t [AlertingType](#) [20]

## 8.1118.1 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[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>AlertingType[VOICE_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>

## 8.1118.2 Field Documentation

8.1118.2.1 `uint8_t voice_arrAlertingType::AlertingType[20]`8.1118.2.2 `uint8_t voice_arrAlertingType::callID[20]`8.1118.2.3 `uint8_t voice_arrAlertingType::numInstances`8.1119 `voice_arrAlphaID` Struct Reference

## Data Fields

- `uint8_t numInstances`
- `voice_allCallsAlphaIDInfo allCallsAlphaIDInfoArr [20]`

## 8.1119.1 Detailed Description

This structure contains an array of Alpha ID Info

## Parameters

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

## 8.1119.2 Field Documentation

8.1119.2.1 `voice_allCallsAlphaIDInfo voice_arrAlphaID::allCallsAlphaIDInfoArr[20]`8.1119.2.2 `uint8_t voice_arrAlphaID::numInstances`8.1120 `voice_arrCalledPartyNum` Struct Reference

## Data Fields

- [uint8\\_t numInstances](#)
- [voice\\_peerNumberInfo](#) [CalledPartyNum](#) [20]

### 8.1120.1 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 <a href="#">calledPartyNum</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>CalledParty-Num[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of <a href="#">CalledPartyNum</a>.</li> <li>• See <a href="#">voice_peerNumberInfo</a> for more information.</li> </ul>

### 8.1120.2 Field Documentation

8.1120.2.1 [voice\\_peerNumberInfo](#) [voice\\_arrCalledPartyNum::CalledPartyNum](#)[20]

8.1120.2.2 [uint8\\_t](#) [voice\\_arrCalledPartyNum::numInstances](#)

## 8.1121 voice\_arrCallEndReason Struct Reference

## Data Fields

- [uint8\\_t numInstances](#)
- [uint8\\_t callID](#) [20]
- [uint16\\_t callEndReason](#) [20]

### 8.1121.1 Detailed Description

This structure contains an array of Call End Reasons.

#### Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of <a href="#">callID</a>, <a href="#">callEndReason</a> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>callID[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>callEndReason[VOICE_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>

### 8.1121.2 Field Documentation

8.1121.2.1 `uint16_t voice_arrCallEndReason::callEndReason[20]`

8.1121.2.2 `uint8_t voice_arrCallEndReason::callID[20]`

8.1121.2.3 `uint8_t voice_arrCallEndReason::numInstances`

## 8.1122 `voice_arrCallInfo` Struct Reference

### Data Fields

- `uint8_t numInstances`
- `voice_getAllCallInformation getAllCallInfo` [20]

### 8.1122.1 Detailed Description

This structure contains an array of Call Info

#### Parameters

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

### 8.1122.2 Field Documentation

8.1122.2.1 `voice_getAllCallInformation voice_arrCallInfo::getAllCallInfo[20]`

8.1122.2.2 `uint8_t voice_arrCallInfo::numInstances`

## 8.1123 `voice_arrConnectPartyNum` Struct Reference

### Data Fields

- `uint8_t numInstances`
- `voice_peerNumberInfo ConnectedPartyNum` [20]

### 8.1123.1 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 <code>ConnectedPartyNum</code> that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
---------------------	---



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

### 8.1123.2 Field Documentation

8.1123.2.1 [voice\\_peerNumberInfo](#) voice\_arrConnectPartyNum::ConnectedPartyNum[20]

8.1123.2.2 [uint8\\_t](#) voice\_arrConnectPartyNum::numInstances

## 8.1124 voice\_arrDiagInfo Struct Reference

### Data Fields

- [uint8\\_t](#) [numInstances](#)
- [voice\\_allCallsDiagInfo](#) [DiagInfo](#) [20]

### 8.1124.1 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> [VOICE_MAX_NO_OF_CALLS]	<ul style="list-style-type: none"> <li>• Array of DiagInfo.</li> <li>• See <a href="#">voice_allCallsDiagInfo</a> for more information.</li> </ul>

### 8.1124.2 Field Documentation

8.1124.2.1 [voice\\_allCallsDiagInfo](#) voice\_arrDiagInfo::DiagInfo[20]

8.1124.2.2 [uint8\\_t](#) voice\_arrDiagInfo::numInstances

## 8.1125 voice\_arrRedirPartyNum Struct Reference

### Data Fields

- [uint8\\_t](#) [numInstances](#)
- [voice\\_peerNumberInfo](#) [RedirPartyNum](#) [20]

### 8.1125.1 Detailed Description

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

## Parameters

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

## 8.1125.2 Field Documentation

8.1125.2.1 `uint8_t voice_arrRedirPartyNum::numInstances`8.1125.2.2 `voice_peerNumberInfo voice_arrRedirPartyNum::RedirPartyNum[20]`8.1126 `voice_arrRemotePartyName` Struct Reference

## Data Fields

- `uint8_t numInstances`
- `voice_getAllCallRmtPtyName GetAllCallRmtPtyName [20]`

## 8.1126.1 Detailed Description

This structure contains an array of Remote Party Names

## Parameters

<i>numInstances</i>	<ul style="list-style-type: none"> <li>• Number of remotePartyName 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 remotePartyName.</li> <li>• See <a href="#">voice_getAllCallRmtPtyName</a> for more information.</li> </ul>

## 8.1126.2 Field Documentation

8.1126.2.1 `voice_getAllCallRmtPtyName voice_arrRemotePartyName::GetAllCallRmtPtyName[20]`8.1126.2.2 `uint8_t voice_arrRemotePartyName::numInstances`8.1127 `voice_arrRemotePartyNum` Struct Reference

## Data Fields

- `uint8_t numInstances`
- `voice_getAllCallRmtPtyNum RmtPtyNum [20]`

### 8.1127.1 Detailed Description

This structure contains an array of Remote Party Numbers

#### Parameters

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

### 8.1127.2 Field Documentation

8.1127.2.1 `uint8_t voice_arrRemotePartyNum::numInstances`

8.1127.2.2 `voice_getAllCallRmtPtyNum voice_arrRemotePartyNum::RmtPtyNum[20]`

## 8.1128 voice\_arrSvcOption Struct Reference

#### Data Fields

- `uint8_t numInstances`
- `uint8_t callID [20]`
- `uint16_t srvOption [20]`

### 8.1128.1 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[VOICE_MAX_NO_OF_CALLS]</i>	<ul style="list-style-type: none"> <li>• Array of Unique call identifier for the call.</li> </ul>
<i>srvOption[VOICE_MAX_NO_OF_CALLS]</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>

### 8.1128.2 Field Documentation

8.1128.2.1 `uint8_t voice_arrSvcOption::callID[20]`

8.1128.2.2 `uint8_t voice_arrSvcOption::numInstances`

8.1128.2.3 `uint16_t voice_arrSvcOption::srvOption[20]`

## 8.1129 `voice_arrUUSInfo` Struct Reference

### Data Fields

- `uint8_t numInstances`
- `voice_allCallsUUSInfo AllCallsUUSInfo [20]`

### 8.1129.1 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 allCallsUUSInfo that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>AllCallsUUS-Info[VOICE_MA-X_NO_OF_CAL-LS]</i>	<ul style="list-style-type: none"> <li>• Array of allCallsUUSInfo.</li> <li>• See <a href="#">voice_allCallsUUSInfo</a> for more information.</li> </ul>

### 8.1129.2 Field Documentation

8.1129.2.1 `voice_allCallsUUSInfo voice_arrUUSInfo::AllCallsUUSInfo[20]`

8.1129.2.2 `uint8_t voice_arrUUSInfo::numInstances`

## 8.1130 `voice_burstDTMFInfo` Struct Reference

### Data Fields

- `uint8_t * pCallID`
- `uint8_t digitCnt`
- `uint8_t pDigitBuff [255]`

### 8.1130.1 Detailed Description

This structure contains Voice Burst DTMF Information

#### Parameters

<i>pCallID</i>	<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> <li>• NULL pointer - Invalid data.</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_VOICE_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>DTMF digit buffer in ASCII, NULL terminated</li> </ul>

## 8.1130.2 Field Documentation

8.1130.2.1 `uint8_t voice_burstDTMFInfo::digitCnt`

8.1130.2.2 `uint8_t* voice_burstDTMFInfo::pCallID`

8.1130.2.3 `uint8_t voice_burstDTMFInfo::pDigitBuff[255]`

## 8.1131 voice\_calledPartyInfo Struct Reference

### Data Fields

- `uint8_t PI`
- `uint8_t SI`
- `uint8_t numType`
- `uint8_t numPlan`
- `uint8_t numLen`
- `uint8_t number` [255]

### 8.1131.1 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>

## 8.1131.2 Field Documentation

8.1131.2.1 `uint8_t voice_calledPartyInfo::number[255]`

8.1131.2.2 `uint8_t voice_calledPartyInfo::numLen`

8.1131.2.3 `uint8_t voice_calledPartyInfo::numPlan`

8.1131.2.4 `uint8_t voice_calledPartyInfo::numType`

8.1131.2.5 `uint8_t voice_calledPartyInfo::PI`

8.1131.2.6 `uint8_t voice_calledPartyInfo::SI`

## 8.1132 voice\_calledPartySubAdd Struct Reference

### Data Fields

- `uint8_t extBit`

- uint8\_t [subAddrType](#)
- uint8\_t [oddEvenInd](#)
- uint8\_t [subAddrLen](#)
- uint8\_t [subAddr](#) [255]

### 8.1132.1 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_VOICE_DESCRIPTION_LENGTH]	<ul style="list-style-type: none"> <li>• Array of the SubAddress in BCD number format.</li> </ul>

### 8.1132.2 Field Documentation

8.1132.2.1 uint8\_t voice\_calledPartySubAdd::extBit

8.1132.2.2 uint8\_t voice\_calledPartySubAdd::oddEvenInd

8.1132.2.3 uint8\_t voice\_calledPartySubAdd::subAddr[255]

8.1132.2.4 uint8\_t voice\_calledPartySubAdd::subAddrLen

8.1132.2.5 uint8\_t voice\_calledPartySubAdd::subAddrType

## 8.1133 voice\_callerIDInfo Struct Reference

#### Data Fields

- uint8\_t [PI](#)
- uint8\_t [callerIDLen](#)
- uint8\_t [callerID](#) [255]

### 8.1133.1 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>

### 8.1133.2 Field Documentation

8.1133.2.1 `uint8_t voice_callerIDInfo::callerID[255]`

8.1133.2.2 `uint8_t voice_callerIDInfo::callerIDLen`

8.1133.2.3 `uint8_t voice_callerIDInfo::PI`

## 8.1134 voice\_callFwdTypeAndPlan Struct Reference

#### Data Fields

- `uint8_t numberType`
- `uint8_t numberPlan`

### 8.1134.1 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>
-------------------	---

## 8.1134.2 Field Documentation

8.1134.2.1 `uint8_t voice_callFwdTypeAndPlan::numberPlan`

8.1134.2.2 `uint8_t voice_callFwdTypeAndPlan::numberType`

## 8.1135 voice\_callFWExtInfo Struct Reference

### Data Fields

- `uint8_t SvcStatus`
- `uint8_t SvcClass`
- `uint8_t noReplyTimer`
- `uint8_t PI`
- `uint8_t SI`
- `uint8_t numType`
- `uint8_t numPlan`
- `uint8_t numLen`
- `uint8_t number` [255]

### 8.1135.1 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>

## 8.1135.2 Field Documentation

8.1135.2.1 `uint8_t voice_callFWExtInfo::noReplyTimer`

8.1135.2.2 `uint8_t voice_callFWExtInfo::number[255]`

8.1135.2.3 uint8\_t voice\_callFWExtInfo::numLen

8.1135.2.4 uint8\_t voice\_callFWExtInfo::numPlan

8.1135.2.5 uint8\_t voice\_callFWExtInfo::numType

8.1135.2.6 uint8\_t voice\_callFWExtInfo::PI

8.1135.2.7 uint8\_t voice\_callFWExtInfo::SI

8.1135.2.8 uint8\_t voice\_callFWExtInfo::SvcClass

8.1135.2.9 uint8\_t voice\_callFWExtInfo::SvcStatus

## 8.1136 voice\_callFWInfo Struct Reference

### Data Fields

- uint8\_t [SvcStatus](#)
- uint8\_t [SvcClass](#)
- uint8\_t [numLen](#)
- uint8\_t [number](#) [255]
- uint8\_t [noReplyTimer](#)

### 8.1136.1 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</i> [255]	<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>

### 8.1136.2 Field Documentation

8.1136.2.1 uint8\_t voice\_callFWInfo::noReplyTimer

8.1136.2.2 `uint8_t voice_callFWInfo::number[255]`

8.1136.2.3 `uint8_t voice_callFWInfo::numLen`

8.1136.2.4 `uint8_t voice_callFWInfo::SvcClass`

8.1136.2.5 `uint8_t voice_callFWInfo::SvcStatus`

## 8.1137 `voice_callInfo` Struct Reference

### Data Fields

- `uint8_t callID`
- `uint8_t callState`
- `uint8_t callType`
- `uint8_t direction`
- `uint8_t mode`

### 8.1137.1 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>

## 8.1137.2 Field Documentation

8.1137.2.1 `uint8_t voice_callInfo::callID`

8.1137.2.2 `uint8_t voice_callInfo::callState`

8.1137.2.3 `uint8_t voice_callInfo::callType`

8.1137.2.4 `uint8_t voice_callInfo::direction`

8.1137.2.5 `uint8_t voice_callInfo::mode`

## 8.1138 voice\_callingPartyInfo Struct Reference

### Data Fields

- `uint8_t PI`
- `uint8_t SI`
- `uint8_t numType`
- `uint8_t numPlan`
- `uint8_t numLen`
- `uint8_t number` [255]

### 8.1138.1 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>
--------------------	---

## 8.1138.2 Field Documentation

8.1138.2.1 uint8\_t voice\_callingPartyInfo::number[255]

8.1138.2.2 uint8\_t voice\_callingPartyInfo::numLen

8.1138.2.3 uint8\_t voice\_callingPartyInfo::numPlan

8.1138.2.4 uint8\_t voice\_callingPartyInfo::numType

8.1138.2.5 uint8\_t voice\_callingPartyInfo::PI

8.1138.2.6 uint8\_t voice\_callingPartyInfo::SI

## 8.1139 voice\_ccSUPSType Struct Reference

### Data Fields

- uint8\_t [svcType](#)
- uint8\_t [reason](#)

### 8.1139.1 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>

## 8.1139.2 Field Documentation

8.1139.2.1 `uint8_t voice_ccSUPSType::reason`

8.1139.2.2 `uint8_t voice_ccSUPSType::svcType`

## 8.1140 `voice_CLIPResp` Struct Reference

### Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

### 8.1140.1 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>

### 8.1140.2 Field Documentation

8.1140.2.1 `uint8_t voice_CLIPResp::ActiveStatus`

8.1140.2.2 `uint8_t voice_CLIPResp::ProvisionStatus`

## 8.1141 `voice_CLIRResp` Struct Reference

### Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

### 8.1141.1 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>

## 8.1141.2 Field Documentation

8.1141.2.1 uint8\_t voice\_CLIRResp::ActiveStatus

8.1141.2.2 uint8\_t voice\_CLIRResp::ProvisionStatus

## 8.1142 voice\_CNAPResp Struct Reference

## Data Fields

- uint8\_t [ActiveStatus](#)
- uint8\_t [ProvisionStatus](#)

## 8.1142.1 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>

### 8.1142.2 Field Documentation

8.1142.2.1 `uint8_t voice_CNAPResp::ActiveStatus`

8.1142.2.2 `uint8_t voice_CNAPResp::ProvisionStatus`

## 8.1143 voice\_COLPResp Struct Reference

### Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

### 8.1143.1 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>

### 8.1143.2 Field Documentation

8.1143.2.1 `uint8_t voice_COLPResp::ActiveStatus`

8.1143.2.2 `uint8_t voice_COLPResp::ProvisionStatus`

## 8.1144 voice\_COLRResp Struct Reference

### Data Fields

- `uint8_t` [ActiveStatus](#)
- `uint8_t` [ProvisionStatus](#)

### 8.1144.1 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>

## 8.1144.2 Field Documentation

8.1144.2.1 uint8\_t voice\_COLRResp::ActiveStatus

8.1144.2.2 uint8\_t voice\_COLRResp::ProvisionStatus

## 8.1145 voice\_connectNumInfo Struct Reference

## Data Fields

- uint8\_t numPresInd
- uint8\_t screeningInd
- uint8\_t numType
- uint8\_t numPlan
- uint8\_t callerIDLen
- uint8\_t callerID [81]

## 8.1145.1 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_VOICE_CALL_NO_LEN]</i>	<ul style="list-style-type: none"> <li>callerID of numLen length, NULL terminated.</li> </ul>

## 8.1145.2 Field Documentation

8.1145.2.1 `uint8_t voice_connectNumInfo::callerID[81]`

8.1145.2.2 `uint8_t voice_connectNumInfo::callerIDLen`

8.1145.2.3 `uint8_t voice_connectNumInfo::numPlan`

8.1145.2.4 uint8\_t voice\_connectNumInfo::numPresInd

8.1145.2.5 uint8\_t voice\_connectNumInfo::numType

8.1145.2.6 uint8\_t voice\_connectNumInfo::screeningInd

## 8.1146 voice\_CUGInfo Struct Reference

### Data Fields

- uint16\_t [CUGIndex](#)
- uint8\_t [SuppPrefCUG](#)
- uint8\_t [SuppOA](#)

### 8.1146.1 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>

### 8.1146.2 Field Documentation

8.1146.2.1 uint16\_t voice\_CUGInfo::CUGIndex

8.1146.2.2 uint8\_t voice\_CUGInfo::SuppOA

8.1146.2.3 uint8\_t voice\_CUGInfo::SuppPrefCUG

## 8.1147 voice\_curAMRConfig Struct Reference

### Data Fields

- uint8\_t [gsmAmrStat](#)
- uint8\_t [wcdmaAmrStat](#)

### 8.1147.1 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>

## 8.1147.2 Field Documentation

8.1147.2.1 `uint8_t voice_curAMRConfig::gsmAmrStat`8.1147.2.2 `uint8_t voice_curAMRConfig::wcdmaAmrStat`8.1148 `voice_diagInfo` Struct Reference

## Data Fields

- `uint8_t diagInfoLen`
- `uint8_t diagnosticInfo [255]`

## 8.1148.1 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_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>Diagnostic information.</li> </ul>

## 8.1148.2 Field Documentation

8.1148.2.1 `uint8_t voice_diagInfo::diagInfoLen`8.1148.2.2 `uint8_t voice_diagInfo::diagnosticInfo[255]`

## 8.1149 voice\_DTMFInfo Struct Reference

### Data Fields

- uint8\_t [callID](#)
- uint8\_t [DTMFEvent](#)
- uint8\_t [digitCnt](#)
- uint8\_t [digitBuff](#) [255]

### 8.1149.1 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_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• DTMF digit buffer in ASCII string which is NULL terminated</li> </ul>

### 8.1149.2 Field Documentation

8.1149.2.1 uint8\_t voice\_DTMFInfo::callID

8.1149.2.2 uint8\_t voice\_DTMFInfo::digitBuff[255]

8.1149.2.3 uint8\_t voice\_DTMFInfo::digitCnt

8.1149.2.4 uint8\_t voice\_DTMFInfo::DTMFEvent

## 8.1150 voice\_DTMFLengths Struct Reference

### Data Fields

- uint8\_t [DTMFPulseWidth](#)
- uint8\_t [DTMFInterdigitInterval](#)

### 8.1150.1 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>

### 8.1150.2 Field Documentation

8.1150.2.1 uint8\_t voice\_DTMFLengths::DTMFInterdigitInterval

8.1150.2.2 uint8\_t voice\_DTMFLengths::DTMFPulseWidth

## 8.1151 voice\_ECTNum Struct Reference

#### Data Fields

- uint8\_t [ECTCallState](#)
- uint8\_t [presentationInd](#)
- uint8\_t [number](#) [81]

### 8.1151.1 Detailed Description

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

#### Parameters

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



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

### 8.1151.2 Field Documentation

8.1151.2.1 `uint8_t voice_ECTNum::ECTCallState`

8.1151.2.2 `uint8_t voice_ECTNum::number[81]`

8.1151.2.3 `uint8_t voice_ECTNum::presentationInd`

## 8.1152 voice\_extDispRecInfo Struct Reference

### Data Fields

- `uint8_t dispType`
- `uint8_t extDispInfoLen`
- `uint8_t extDispInfo [255]`

### 8.1152.1 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>

### 8.1152.2 Field Documentation

8.1152.2.1 `uint8_t voice_extDispRecInfo::dispType`

8.1152.2.2 `uint8_t voice_extDispRecInfo::extDispInfo[255]`

8.1152.2.3 `uint8_t voice_extDispRecInfo::extDispInfoLen`

## 8.1153 `voice_getAllCallInformation` Struct Reference

### Data Fields

- [voice\\_callInfo](#) `Callinfo`
- `uint8_t isEmpty`
- `uint8_t ALS`

### 8.1153.1 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="#">voice_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>

### 8.1153.2 Field Documentation

8.1153.2.1 `uint8_t voice_getAllCallInformation::ALS`

8.1153.2.2 `voice_callInfo voice_getAllCallInformation::Callinfo`

8.1153.2.3 `uint8_t voice_getAllCallInformation::isEmpty`

## 8.1154 `voice_getAllCallRmtPtyName` Struct Reference

### Data Fields

- `uint8_t callID`
- [voice\\_remotePartyName](#) `RemotePartyName`

### 8.1154.1 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="#">voice_remotePartyName</a> for more information.</li> </ul>

## 8.1154.2 Field Documentation

8.1154.2.1 uint8\_t voice\_getAllCallRmtPtyName::callID

8.1154.2.2 voice\_remotePartyName voice\_getAllCallRmtPtyName::RemotePartyName

## 8.1155 voice\_getAllCallRmtPtyNum Struct Reference

## Data Fields

- uint8\_t [callID](#)
- [voice\\_remotePartyNum RemotePartyNum](#)

## 8.1155.1 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="#">voice_remotePartyNum</a> for more information.</li> </ul>

## 8.1155.2 Field Documentation

8.1155.2.1 uint8\_t voice\_getAllCallRmtPtyNum::callID

8.1155.2.2 voice\_remotePartyNum voice\_getAllCallRmtPtyNum::RemotePartyNum

## 8.1156 voice\_getCallFWExtInfo Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- [voice\\_callFWExtInfo CallFWExtInfo](#) [20]

## 8.1156.1 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 callFWExtInfo that follow.</li> <li>• If zero(0) then no further information exists.</li> </ul>
<i>CallFWExtInfo[VOICE_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="#">voice_callFWExtInfo</a> for more information.</li> </ul> </li> </ul>

## 8.1156.2 Field Documentation

8.1156.2.1 `voice_callFWExtInfo voice_getCallFWExtInfo::CallFWExtInfo[20]`8.1156.2.2 `uint8_t voice_getCallFWExtInfo::numInstances`8.1157 `voice_getCallFWInfo` Struct Reference

## Data Fields

- `uint8_t numInstances`
- `voice_callFWInfo CallFWInfo [20]`

## 8.1157.1 Detailed Description

This structure contains an array of Call Forwarded Information.

## Parameters

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

## 8.1157.2 Field Documentation

8.1157.2.1 `voice_callFWInfo voice_getCallFWInfo::CallFWInfo[20]`8.1157.2.2 `uint8_t voice_getCallFWInfo::numInstances`8.1158 `voice_lineCtrlInfo` Struct Reference

## Data Fields

- `uint8_t polarityIncluded`
- `uint8_t toggleMode`
- `uint8_t revPolarity`
- `uint8_t pwrDenialTime`

### 8.1158.1 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>

### 8.1158.2 Field Documentation

8.1158.2.1 uint8\_t voice\_lineCtrlInfo::polarityIncluded

8.1158.2.2 uint8\_t voice\_lineCtrlInfo::pwrDenialTime

8.1158.2.3 uint8\_t voice\_lineCtrlInfo::revPolarity

8.1158.2.4 uint8\_t voice\_lineCtrlInfo::toggleMode

## 8.1159 voice\_newPwdData Struct Reference

### Data Fields

- uint8\_t [newPwd](#) [4]
- uint8\_t [newPwdAgain](#) [4]

### 8.1159.1 Detailed Description

This structure contains New Password Data.

#### Parameters

<i>newPwd[BARRING_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>newPwdAgain[BARRING_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>

## 8.1159.2 Field Documentation

8.1159.2.1 `uint8_t voice_newPwdData::newPwd[4]`

8.1159.2.2 `uint8_t voice_newPwdData::newPwdAgain[4]`

## 8.1160 voice\_NSSAudioCtrl Struct Reference

### Data Fields

- `uint8_t upLink`
- `uint8_t downLink`

### 8.1160.1 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>

## 8.1160.2 Field Documentation

8.1160.2.1 `uint8_t voice_NSSAudioCtrl::downLink`

8.1160.2.2 `uint8_t voice_NSSAudioCtrl::upLink`

## 8.1161 voice\_peerNumberInfo Struct Reference

### Data Fields

- `uint8_t callID`
- `uint8_t numPI`
- `uint8_t numSI`
- `uint8_t numType`
- `uint8_t numPlan`
- `uint8_t numLen`
- `uint8_t number` [81]

### 8.1161.1 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_VOICE_CALL_NO_LEN]	<ul style="list-style-type: none"> <li>• number of numLen length, NULL terminated.</li> </ul>

## 8.1161.2 Field Documentation

- 8.1161.2.1 `uint8_t voice_peerNumberInfo::callID`
- 8.1161.2.2 `uint8_t voice_peerNumberInfo::number[81]`
- 8.1161.2.3 `uint8_t voice_peerNumberInfo::numLen`
- 8.1161.2.4 `uint8_t voice_peerNumberInfo::numPI`
- 8.1161.2.5 `uint8_t voice_peerNumberInfo::numPlan`
- 8.1161.2.6 `uint8_t voice_peerNumberInfo::numSI`
- 8.1161.2.7 `uint8_t voice_peerNumberInfo::numType`

## 8.1162 `voice_prefVoiceSO` Struct Reference

### Data Fields

- `uint8_t namID`
- `uint8_t evrcCapability`
- `uint16_t homePageVoiceSO`
- `uint16_t homeOrigVoiceSO`
- `uint16_t roamOrigVoiceSO`

### 8.1162.1 Detailed Description

This structure contains information about the Preferred Voice Service Options.

#### Parameters

<i>namID</i>	<ul style="list-style-type: none"> <li>• Index of the NAM(Number Assignment Module) to be configured.</li> <li>• Range 0 to 3.</li> <li>• Some modems support only 1 or 2 NAMs.</li> <li>• 0xFF,if not available.</li> </ul>
<i>evrcCapability</i>	<ul style="list-style-type: none"> <li>• EVRC capability.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x00 - Disable</li> <li>– 0x01 - Enable</li> <li>– 0xFF - Not Available</li> </ul> </li> </ul>



<i>homePageVoice-</i> SO	<ul style="list-style-type: none"> <li>• Home page voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x0000 - VOICE_SO_WILD - Any service option</li> <li>– 0x0001 - VOICE_SO_IS_96A - IS-96A</li> <li>– 0x0003 - VOICE_SO_EVRC - EVRC</li> <li>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733</li> <li>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder</li> <li>– 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband</li> <li>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband</li> <li>– 0x8000 - VOICE_SO_13K - 13K</li> <li>– 0x8001 - VOICE_SO_IS_96 - IS-96</li> <li>– 0x8023 - VOICE_SO_WVRC - WVRC</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>homeOrigVoice-</i> SO	<ul style="list-style-type: none"> <li>• Home origination voice SO; most preferred CDMA SO to be requested from the network when receiving an incoming (MT) voice call within the home network.</li> <li>• Values: <ul style="list-style-type: none"> <li>– 0x0000 - VOICE_SO_WILD - Any service option</li> <li>– 0x0001 - VOICE_SO_IS_96A - IS-96A</li> <li>– 0x0003 - VOICE_SO_EVRC - EVRC</li> <li>– 0x0011 - VOICE_SO_13K_IS733 - 13K_IS733</li> <li>– 0x0038 - VOICE_SO_SELECTABLE_MODE_VOCODER - Selectable mode vocoder</li> <li>– 0x0044 - VOICE_SO_4GV_NARROW_BAND - 4GV narrowband</li> <li>– 0x0046 - VOICE_SO_4GV_WIDE_BAND - 4GV wideband</li> <li>– 0x8000 - VOICE_SO_13K - 13K</li> <li>– 0x8001 - VOICE_SO_IS_96 - IS-96</li> <li>– 0x8023 - VOICE_SO_WVRC - WVRC</li> <li>– 0xFFFF - Not Available</li> </ul> </li> </ul>
<i>roamOrigVoice-</i> SO	<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>

### 8.1162.2 Field Documentation

8.1162.2.1 uint8\_t voice\_prefVoiceSO::evrcCapability

8.1162.2.2 uint16\_t voice\_prefVoiceSO::homeOrigVoiceSO

8.1162.2.3 uint16\_t voice\_prefVoiceSO::homePageVoiceSO

8.1162.2.4 uint8\_t voice\_prefVoiceSO::namID

8.1162.2.5 uint16\_t voice\_prefVoiceSO::roamOrigVoiceSO

## 8.1163 voice\_redirNumInfo Struct Reference

### Data Fields

- uint8\_t [PI](#)
- uint8\_t [SI](#)
- uint8\_t [numType](#)
- uint8\_t [numPlan](#)
- uint8\_t [reason](#)
- uint8\_t [numLen](#)
- uint8\_t [number](#) [255]

### 8.1163.1 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>

## 8.1163.2 Field Documentation

8.1163.2.1 `uint8_t voice_redirNumInfo::number[255]`

8.1163.2.2 `uint8_t voice_redirNumInfo::numLen`

8.1163.2.3 `uint8_t voice_redirNumInfo::numPlan`

8.1163.2.4 `uint8_t voice_redirNumInfo::numType`

8.1163.2.5 `uint8_t voice_redirNumInfo::PI`

8.1163.2.6 `uint8_t voice_redirNumInfo::reason`

8.1163.2.7 `uint8_t voice_redirNumInfo::SI`

## 8.1164 voice\_remotePartyName Struct Reference

### Data Fields

- uint8\_t [namePI](#)
- uint8\_t [codingScheme](#)
- uint8\_t [nameLen](#)
- uint8\_t [callerName](#) [255]

### 8.1164.1 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_VOICE_DESCRIPTION_LENGTH]</i>	<ul style="list-style-type: none"> <li>• Name in ASCII, NULL ending.</li> </ul>

### 8.1164.2 Field Documentation

8.1164.2.1 uint8\_t voice\_remotePartyName::callerName[255]

8.1164.2.2 uint8\_t voice\_remotePartyName::codingScheme

8.1164.2.3 uint8\_t voice\_remotePartyName::nameLen

8.1164.2.4 uint8\_t voice\_remotePartyName::namePI

## 8.1165 voice\_remotePartyNum Struct Reference

## Data Fields

- uint8\_t [presentationInd](#)
- uint8\_t [numLen](#)
- uint8\_t [remPartyNumber](#) [81]

### 8.1165.1 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>

### 8.1165.2 Field Documentation

8.1165.2.1 uint8\_t voice\_remotePartyNum::numLen

8.1165.2.2 uint8\_t voice\_remotePartyNum::presentationInd

8.1165.2.3 uint8\_t voice\_remotePartyNum::remPartyNumber[81]

## 8.1166 voice\_roamTimer Struct Reference

## Data Fields

- uint8\_t [namID](#)
- uint32\_t [roamTimerValue](#)

### 8.1166.1 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>

## 8.1166.2 Field Documentation

8.1166.2.1 uint8\_t voice\_roamTimer::namID

8.1166.2.2 uint32\_t voice\_roamTimer::roamTimerValue

## 8.1167 voice\_signalInfo Struct Reference

## Data Fields

- uint8\_t [signalType](#)
- uint8\_t [alertPitch](#)
- uint8\_t [signal](#)

## 8.1167.1 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>

## 8.1167.2 Field Documentation

8.1167.2.1 uint8\_t voice\_signalInfo::alertPitch

8.1167.2.2 uint8\_t voice\_signalInfo::signal

8.1167.2.3 uint8\_t voice\_signalInfo::signalType

## 8.1168 voice\_SUPSInfo Struct Reference

### Data Fields

- uint8\_t [svcType](#)
- uint8\_t [isModByCC](#)

### 8.1168.1 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>

### 8.1168.2 Field Documentation

8.1168.2.1 uint8\_t voice\_SUPSInfo::isModByCC

8.1168.2.2 uint8\_t voice\_SUPSInfo::svcType

## 8.1169 voice\_USSDNotificationNetworkInfo Struct Reference

### Data Fields

- uint8\_t [tlvPresent](#)
- struct [voice\\_USSInfo networkInfo](#)

### 8.1169.1 Detailed Description

Structure for storing the USS info present in USSDNotification callback.

#### Parameters

<i>tlvPresent</i>	Values: <ul style="list-style-type: none"><li>• 0 - Not present</li><li>• 1 - Tlv is present</li></ul>
-------------------	--

<i>NetworkInfo</i>	<ul style="list-style-type: none"> <li>• USS Data from Network (See structure <a href="#">voice_USSInfo</a>)</li> <li>• NULL pointer - Invalid data.</li> </ul>
--------------------	---

## 8.1169.2 Field Documentation

8.1169.2.1 struct [voice\\_USSInfo](#) [voice\\_USSDNotificationNetworkInfo::networkInfo](#)

8.1169.2.2 uint8\_t [voice\\_USSDNotificationNetworkInfo::tlvPresent](#)

## 8.1170 voice\_USSInfo Struct Reference

### Data Fields

- uint8\_t [ussDCS](#)
- uint8\_t [ussLen](#)
- uint8\_t [ussData](#) [182]

### 8.1170.1 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>

## 8.1170.2 Field Documentation

8.1170.2.1 uint8\_t [voice\\_USSInfo::ussData](#)[182]

8.1170.2.2 uint8\_t [voice\\_USSInfo::ussDCS](#)

8.1170.2.3 uint8\_t [voice\\_USSInfo::ussLen](#)

## 8.1171 voice\_UUSInfo Struct Reference

### Data Fields

- uint8\_t [UUSType](#)
- uint8\_t [UUSDcs](#)



- uint8\_t [UUSDatalen](#)
- uint8\_t [UUSData](#) [255]

### 8.1171.1 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[<a href="#">MAX_VOICE_DESCRIPTION_LENGTH</a>]</i>	<ul style="list-style-type: none"> <li>• UUS data encoded as per coding scheme</li> </ul>

### 8.1171.2 Field Documentation

8.1171.2.1 uint8\_t voice\_UUSInfo::UUSData[255]

8.1171.2.2 uint8\_t voice\_UUSInfo::UUSDatalen

8.1171.2.3 uint8\_t voice\_UUSInfo::UUSDcs

8.1171.2.4 uint8\_t voice\_UUSInfo::UUSType

## 8.1172 wds\_channelRate Struct Reference

### Data Fields

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

### 8.1172.1 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>

### 8.1172.2 Field Documentation

8.1172.2.1 uint32\_t wds\_channelRate::CurrChanRxRate

8.1172.2.2 uint32\_t wds\_channelRate::CurrChanTxRate

## 8.1173 wds\_ChannelRateTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- [wds\\_channelRate](#) [ChannelRate](#)

### 8.1173.1 Detailed Description

This structure stores information about channel rate TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>ChannelRate</i>	<ul style="list-style-type: none"> <li>• see <a href="#">wds_channelRate</a> for more info</li> </ul>

### 8.1173.2 Field Documentation

8.1173.2.1 wds\_channelRate wds\_ChannelRateTlv::ChannelRate

8.1173.2.2 uint8\_t wds\_ChannelRateTlv::TlvPresent

## 8.1174 wds\_ConnStatusTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [MDMConnStatus](#)

### 8.1174.1 Detailed Description

This structure stores information about modem connection status TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>MDMConn-Status</i>	<ul style="list-style-type: none"><li>-Modem connecton status</li><li>• 0x01 - disconnected</li><li>• 0x02 - connected</li></ul>

### 8.1174.2 Field Documentation

8.1174.2.1 uint8\_t wds\_ConnStatusTlv::MDMConnStatus

8.1174.2.2 uint8\_t wds\_ConnStatusTlv::TlvPresent

## 8.1175 wds\_currNetworkInfo Struct Reference

### Data Fields

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

### 8.1175.1 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>RAT mask to indicate type of technology</li> <li>Values <ul style="list-style-type: none"> <li>0x00 - DONT_CARE</li> <li>0x8000 - NULL_BEARER</li> </ul> </li> <li>CDMA RAT mask <ul style="list-style-type: none"> <li>0x01 - CDMA_1X</li> <li>0x02 - EVDO_REV0</li> <li>0x05 - HRPD</li> <li>0x0B - EHRPD</li> </ul> </li> <li>UMTS RAT mask <ul style="list-style-type: none"> <li>0x03 - GPRS</li> <li>0x04 - WCDMA</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>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> </ul> </li> </ul>
<i>SOMask</i>	<ul style="list-style-type: none"> <li><a href="#">SO Mask</a></li> </ul>

## 8.1175.2 Field Documentation

8.1175.2.1 `uint8_t wds_currNetworkInfo::NetworkType`8.1175.2.2 `uint32_t wds_currNetworkInfo::RATMask`8.1175.2.3 `uint32_t wds_currNetworkInfo::SOMask`8.1176 `wds_DataBearerTechTlv` Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `uint8_t DataBearerTech`

## 8.1176.1 Detailed Description

This structure hold parameters about data bearer technology TLV.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>DataBearerTech</i>	<ul style="list-style-type: none"> <li>• Data bear technology <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>

## 8.1176.2 Field Documentation

8.1176.2.1 uint8\_t wds\_DataBearTechTlv::DataBearerTech

8.1176.2.2 uint8\_t wds\_DataBearTechTlv::TlvPresent

## 8.1177 wds\_DataULongLongTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [uIlData](#)

## 8.1177.1 Detailed Description

This structure contains Data unsigned long long TLV data.

## Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>uIlData</i>	unsigend long long data in 64bit

## 8.1177.2 Field Documentation

8.1177.2.1 `uint8_t wds_DataULongLongTlv::TlvPresent`

8.1177.2.2 `uint64_t wds_DataULongLongTlv::ulldata`

## 8.1178 wds\_DataULongTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint32_t ulData`

### 8.1178.1 Detailed Description

This structure contains Data unsigned long TLV data.

#### Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>ulData</i>	unsigend long data in 32bit

## 8.1178.2 Field Documentation

8.1178.2.1 `uint8_t wds_DataULongTlv::TlvPresent`

8.1178.2.2 `uint32_t wds_DataULongTlv::ulData`

## 8.1179 wds\_DHCPLeaseOptTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint8_t numOpt`
- `wds_DHCPOpt optList [30]`
- `uint8_t optListData [2048]`

### 8.1179.1 Detailed Description

DHCP lease option information

#### Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>numOpt</i>	number of sets of <code>wds_DHCPOpt</code>
<i>optList</i>	option list
<i>optListData</i>	option list data

## 8.1179.2 Field Documentation

8.1179.2.1 `uint8_t wds_DHCPLeaseOptTlv::numOpt`

8.1179.2.2 wds\_DHCPOpt wds\_DHCPLeaseOptTlv::optList[30]

8.1179.2.3 uint8\_t wds\_DHCPLeaseOptTlv::optListData[2048]

8.1179.2.4 uint8\_t wds\_DHCPLeaseOptTlv::TlvPresent

## 8.1180 wds\_DHCPLeaseStateTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [leaseState](#)

### 8.1180.1 Detailed Description

DHCP lease state information

#### Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>leaseState</i>	lease state <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>

### 8.1180.2 Field Documentation

8.1180.2.1 uint8\_t wds\_DHCPLeaseStateTlv::leaseState

8.1180.2.2 uint8\_t wds\_DHCPLeaseStateTlv::TlvPresent

## 8.1181 wds\_DHCPOpt Struct Reference

### Data Fields

- uint8\_t [optCode](#)
- uint8\_t [optValLen](#)
- uint8\_t \* [pOptVal](#)

### 8.1181.1 Detailed Description

DHCP option code information

## Parameters

<i>optCode</i>	option code <ul style="list-style-type: none"> <li>values: <ul style="list-style-type: none"> <li>0 - 255</li> </ul> </li> </ul>
<i>optValLen</i>	length of option code <ul style="list-style-type: none"> <li>values: <ul style="list-style-type: none"> <li>0 - 255</li> </ul> </li> </ul>
<i>pOptVal</i>	option value

## 8.1181.2 Field Documentation

8.1181.2.1 `uint8_t wds_DHCPOpt::optCode`8.1181.2.2 `uint8_t wds_DHCPOpt::optValLen`8.1181.2.3 `uint8_t* wds_DHCPOpt::pOptVal`8.1182 `wds_DHCPProfileIdTlv` Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `uint8_t profileType`
- `uint8_t profileId`

## 8.1182.1 Detailed Description

DHCP profile id TLV information

## Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>profileType</i>	identifying the type of the profile <ul style="list-style-type: none"> <li>0 - 3GPP</li> </ul>
<i>profileId</i>	index identifying the profile <ul style="list-style-type: none"> <li>1-24 valid for 3GPP profile type (9x30 and onwards)</li> </ul>

## 8.1182.2 Field Documentation

8.1182.2.1 `uint8_t wds_DHCPProfileIdTlv::profileId`8.1182.2.2 `uint8_t wds_DHCPProfileIdTlv::profileType`8.1182.2.3 `uint8_t wds_DHCPProfileIdTlv::TlvPresent`8.1183 `wds_DHCPv4HWConfig` Struct Reference



## Data Fields

- uint8\_t [hwType](#)
- uint8\_t [chaddrLen](#)
- uint8\_t [chaddr](#) [16]

### 8.1183.1 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>

### 8.1183.2 Field Documentation

8.1183.2.1 uint8\_t wds\_DHCPv4HWConfig::chaddr[16]

8.1183.2.2 uint8\_t wds\_DHCPv4HWConfig::chaddrLen

8.1183.2.3 uint8\_t wds\_DHCPv4HWConfig::hwType

## 8.1184 wds\_DHCPv4Option Struct Reference

## Data Fields

- uint8\_t [optCode](#)
- uint8\_t [optValLen](#)
- uint8\_t [optVal](#) [255]

### 8.1184.1 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>
---------------	--

## 8.1184.2 Field Documentation

8.1184.2.1 `uint8_t wds_DHCPv4Option::optCode`

8.1184.2.2 `uint8_t wds_DHCPv4Option::optVal[255]`

8.1184.2.3 `uint8_t wds_DHCPv4Option::optValLen`

## 8.1185 wds\_DHCPv4OptionList Struct Reference

### Data Fields

- `uint8_t numOpt`
- `wds_DHCPv4Option * pOptList`

### 8.1185.1 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>

## 8.1185.2 Field Documentation

8.1185.2.1 `uint8_t wds_DHCPv4OptionList::numOpt`

8.1185.2.2 `wds_DHCPv4Option* wds_DHCPv4OptionList::pOptList`

## 8.1186 wds\_DHCPv4ProfileId Struct Reference

### Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

### 8.1186.1 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>

## 8.1186.2 Field Documentation

8.1186.2.1 uint8\_t wds\_DHCPv4ProfileId::profileId

8.1186.2.2 uint8\_t wds\_DHCPv4ProfileId::profileType

## 8.1187 wds\_Domain Struct Reference

## Data Fields

- uint16\_t [domainLen](#)
- uint8\_t [domainName](#) [256]

## 8.1187.1 Detailed Description

This structure contains the DomainName Information

## Parameters

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

## 8.1187.2 Field Documentation

8.1187.2.1 uint16\_t wds\_Domain::domainLen

8.1187.2.2 uint8\_t wds\_Domain::domainName[256]

## 8.1188 wds\_DomainNameList Struct Reference

## Data Fields

- uint8\_t [numInstances](#)
- struct [wds\\_Domain domain](#) [10]

## 8.1188.1 Detailed Description

This structure contains the DomainNameList Information

## Parameters

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

## Note

Data invalid when [numInstances](#) equal 0xff.

## 8.1188.2 Field Documentation

8.1188.2.1 struct wds\_Domain wds\_DomainNameList::domain[10]

8.1188.2.2 uint8\_t wds\_DomainNameList::numInstances

## 8.1189 wds\_DormStatTlv Struct Reference

## Data Fields

- uint8\_t [TlvPresent](#)
- uint8\_t [DormancyStat](#)

## 8.1189.1 Detailed Description

This structure stores information about dormancy status TLV.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>DormancyStat</i>	-Dormancy status <ul style="list-style-type: none"> <li>• 0x01 - Traffic channel dormant</li> <li>• 0x02 - Traffic channel active</li> </ul>

## 8.1189.2 Field Documentation

8.1189.2.1 uint8\_t wds\_DormStatTlv::DormancyStat

8.1189.2.2 uint8\_t wds\_DormStatTlv::TlvPresent

## 8.1190 wds\_GPRSQoS Struct Reference

## Data Fields

- uint32\_t [precedenceClass](#)
- uint32\_t [delayClass](#)
- uint32\_t [reliabilityClass](#)

- uint32\_t [peakThroughputClass](#)
- uint32\_t [meanThroughputClass](#)

### 8.1190.1 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>

#### Note

Data invalid when all parameters are equal to 0xffffffff.

### 8.1190.2 Field Documentation

8.1190.2.1 uint32\_t wds\_GPRSQoS::delayClass

8.1190.2.2 uint32\_t wds\_GPRSQoS::meanThroughputClass

8.1190.2.3 uint32\_t wds\_GPRSQoS::peakThroughputClass

8.1190.2.4 uint32\_t wds\_GPRSQoS::precedenceClass

8.1190.2.5 uint32\_t wds\_GPRSQoS::reliabilityClass

## 8.1191 wds\_IPv4AdTlv Struct Reference

#### Data Fields

- uint8\_t [TlvPresent](#)
- uint32\_t [IPv4Addr](#)

### 8.1191.1 Detailed Description

IPv4 address TLV information

## Parameters

<i>TlvPresent</i>	a flag indicates whether TLV presented or not
<i>IPv4Addr</i>	IPv4 address

## 8.1191.2 Field Documentation

8.1191.2.1 uint32\_t wds\_IPv4AdTlv::IPv4Addr

8.1191.2.2 uint8\_t wds\_IPv4AdTlv::TlvPresent

## 8.1192 wds\_IPV6AddressInfo Struct Reference

## Data Fields

- uint8\_t [IPV6PrefixLen](#)
- uint16\_t [IPAddressV6](#) [8]

## 8.1192.1 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>

## 8.1192.2 Field Documentation

8.1192.2.1 uint16\_t wds\_IPV6AddressInfo::IPAddressV6[8]

8.1192.2.2 uint8\_t wds\_IPV6AddressInfo::IPV6PrefixLen

## 8.1193 wds\_IPV6GWAddressInfo Struct Reference

## Data Fields

- uint8\_t [gwV6PrefixLen](#)
- uint16\_t [gwAddressV6](#) [8]

## 8.1193.1 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>

## Note

Data invalid when IPV6PrefixLen equal 0xff.

## 8.1193.2 Field Documentation

8.1193.2.1 `uint16_t wds_IPV6GWAddressInfo::gwAddressV6[8]`

8.1193.2.2 `uint8_t wds_IPV6GWAddressInfo::gwV6PrefixLen`

## 8.1194 wds\_LastMdmCallEndRsnTlv Struct Reference

## Data Fields

- `uint8_t TlvPresent`
- `uint16_t CallEndReason`

## 8.1194.1 Detailed Description

This structure stores information about modem last call end reason TLV.

## Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>CallEndReason</i>	-Modem call end reason <ul style="list-style-type: none"> <li>See <a href="#">qaGobiApiTableCallEndReasons.h</a> for Call End Reason</li> </ul>

## 8.1194.2 Field Documentation

8.1194.2.1 `uint16_t wds_LastMdmCallEndRsnTlv::CallEndReason`

8.1194.2.2 `uint8_t wds_LastMdmCallEndRsnTlv::TlvPresent`

## 8.1195 wds\_PCSCFFQDNAddress Struct Reference

## Data Fields

- `uint16_t fqdnLen`
- `uint8_t fqdnAddr [256]`

### 8.1195.1 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>

### 8.1195.2 Field Documentation

8.1195.2.1 `uint8_t wds_PCSCFFQDNAddress::fqdnAddr[256]`

8.1195.2.2 `uint16_t wds_PCSCFFQDNAddress::fqdnLen`

## 8.1196 wds\_PCSCFFQDNAddressList Struct Reference

### Data Fields

- `uint8_t numInstances`
- struct `wds_PCSCFFQDNAddress pcsfFQDNAddress` [10]

### 8.1196.1 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>

#### Note

Data invalid when `numInstances` equal 0xff.

### 8.1196.2 Field Documentation

8.1196.2.1 `uint8_t wds_PCSCFFQDNAddressList::numInstances`

8.1196.2.2 `struct wds_PCSCFFQDNAddress wds_PCSCFFQDNAddressList::pcsfFQDNAddress[10]`

## 8.1197 wds\_PCSCFIPv4ServerAddressList Struct Reference

### Data Fields

- `uint8_t numInstances`



- uint32\_t [pscsfIPV4Addr](#) [64]

### 8.1197.1 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>pscsfIPV4Addr</i>	<ul style="list-style-type: none"> <li>• P-CSCF IPv4 server addresses(Max 16 address, 4 bytes each)</li> </ul>

#### Note

Data invalid when [numInstances](#) equal 0xff and [pscsfIPV4Addr](#) equal to 0xffffffff.

### 8.1197.2 Field Documentation

8.1197.2.1 uint8\_t wds\_PCSCFIPv4ServerAddressList::numInstances

8.1197.2.2 uint32\_t wds\_PCSCFIPv4ServerAddressList::pscsfIPV4Addr[64]

## 8.1198 wds\_profileChange Struct Reference

### Data Fields

- uint8\_t [profileType](#)
- uint8\_t [profileIdx](#)

### 8.1198.1 Detailed Description

This structure contains Profile Change Info

#### Parameters

<i>profileType</i>	<ul style="list-style-type: none"> <li>• Identifies the type of the profile</li> </ul>
<i>profileIdx</i>	<ul style="list-style-type: none"> <li>• Index identifying the profile</li> </ul>

### 8.1198.2 Field Documentation

8.1198.2.1 uint8\_t wds\_profileChange::profileIdx

8.1198.2.2 uint8\_t wds\_profileChange::profileType

## 8.1199 wds\_ProfileIdentifier Struct Reference

## Data Fields

- uint8\_t [profileType](#)
- uint8\_t [profileIndex](#)

### 8.1199.1 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>

#### Note

Data invalid when all parameters are equal to 0xff.

### 8.1199.2 Field Documentation

8.1199.2.1 uint8\_t wds\_ProfileIdentifier::profileIndex

8.1199.2.2 uint8\_t wds\_ProfileIdentifier::profileType

## 8.1200 wds\_profileInfo Union Reference

## Data Fields

- [LibPackprofile\\_3GPP](#) SlqsProfile3GPP
- [LibPackprofile\\_3GPP2](#) SlqsProfile3GPP2

### 8.1200.1 Detailed Description

This union consist of profile\_3GPP and profile\_3GPP2 out of which one will be used to create profile.

#### Parameters

<i>SlqsProfile3GPP</i>	3GPP profile See <a href="#">LibPackprofile_3GPP</a>
<i>SlqsProfile3GPP2</i>	3GPP2 profile See <a href="#">LibPackprofile_3GPP2</a>

### 8.1200.2 Field Documentation

8.1200.2.1 [LibPackprofile\\_3GPP](#) wds\_profileInfo::SlqsProfile3GPP

8.1200.2.2 [LibPackprofile\\_3GPP2](#) wds\_profileInfo::SlqsProfile3GPP2

## 8.1201 wds\_RXBytesOKTlv Struct Reference

### Data Fields

- uint8\_t [TlvPresent](#)
- uint64\_t [RxByteOKCnt](#)

### 8.1201.1 Detailed Description

This structure stores information about Rx Bytes OK TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"><li>• value is 1 if this TLV is present in indication, otherwise 0</li></ul>
<i>RxByteOKCnt</i>	-Number of bytes received without error

### 8.1201.2 Field Documentation

8.1201.2.1 uint64\_t wds\_RXBytesOKTlv::RxByteOKCnt

8.1201.2.2 uint8\_t wds\_RXBytesOKTlv::TlvPresent

## 8.1202 wds\_sourceOfChange Struct Reference

### Data Fields

- uint8\_t [source](#)

### 8.1202.1 Detailed Description

This structure contains Source of change Info

#### Parameters

<i>source</i>	<ul style="list-style-type: none"><li>• source of change</li></ul>
---------------	--

### 8.1202.2 Field Documentation

8.1202.2.1 uint8\_t wds\_sourceOfChange::source

## 8.1203 wds\_transferStatInd Struct Reference

### Data Fields

- uint8\_t [StatsPeriod](#)
- uint32\_t [StatsMask](#)

### 8.1203.1 Detailed Description

This structure contains the transfer statistic indication information

#### Parameters

<i>StatsPeriod</i>	Field Period between transfer statistic reports. <ul style="list-style-type: none"> <li>• 0 - Do not report.</li> <li>• 1 - Other - Period between reports (in seconds).</li> </ul>
<i>StatsMask</i>	requested statistic bit mask. <ul style="list-style-type: none"> <li>• 0x00000040 - Tx bytes OK</li> <li>• 0x00000080 - Rx bytes OK</li> </ul>

### 8.1203.2 Field Documentation

8.1203.2.1 `uint32_t wds_transferStatInd::StatsMask`

8.1203.2.2 `uint8_t wds_transferStatInd::StatsPeriod`

## 8.1204 wds\_TrStatInd Struct Reference

### Data Fields

- `uint8_t statsPeriod`
- `uint32_t statsMask`

### 8.1204.1 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 <code>SLQSWdsEventReport-Callback</code>.</li> <li>• All unlisted bits are reserved for future use and must be set to zero.</li> </ul>

## 8.1204.2 Field Documentation

8.1204.2.1 `uint32_t wds_TrStatInd::statsMask`

8.1204.2.2 `uint8_t wds_TrStatInd::statsPeriod`

## 8.1205 wds\_TXBytesOKTlv Struct Reference

### Data Fields

- `uint8_t TlvPresent`
- `uint64_t TxByteOKCnt`

### 8.1205.1 Detailed Description

This structure stores information about Tx Bytes OK TLV.

#### Parameters

<i>TlvPresent</i>	<ul style="list-style-type: none"> <li>• value is 1 if this TLV is present in indication, otherwise 0</li> </ul>
<i>TxByteOKCnt</i>	<ul style="list-style-type: none"> <li>• Number of bytes transmitted without error</li> </ul>

## 8.1205.2 Field Documentation

8.1205.2.1 `uint8_t wds_TXBytesOKTlv::TlvPresent`

8.1205.2.2 `uint64_t wds_TXBytesOKTlv::TxByteOKCnt`

## 8.1206 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`

### 8.1206.1 Detailed Description

This structure contains the UMTS Quality Of Service Information

## Parameters

<i>trafficClass</i>	<ul style="list-style-type: none"> <li>• 0x00 - Subscribed</li> <li>• 0x01 - Conversational</li> <li>• 0x02 - Streaming</li> <li>• 0x03 - Interactive</li> <li>• 0x04 - Background</li> </ul>
<i>maxUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Maximum uplink bit rate in bits/sec</li> </ul>
<i>maxDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Maximum downlink bit rate in bits/sec</li> </ul>
<i>grntUplinkBitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed uplink bit rate in bits/sec</li> </ul>
<i>grntDownlink-Bitrate</i>	<ul style="list-style-type: none"> <li>• Guaranteed downlink bit rate in bits/sec</li> </ul>
<i>qosDelivery-Order</i>	<ul style="list-style-type: none"> <li>- Qos delivery order</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - Delivery order on</li> <li>• 0x02 - Delivery order off</li> </ul>
<i>maxSDUSize</i>	<ul style="list-style-type: none"> <li>• Maximum SDU size</li> </ul>
<i>sduErrorRatio</i>	<ul style="list-style-type: none"> <li>- SDU error ratio</li> <li>• Target value for fraction of SDUs lost or detected as erroneous.</li> <li>• 0x00 - Subscribe</li> <li>• 0x01 - <math>1 \times 10^{-2}</math></li> <li>• 0x02 - <math>7 \times 10^{-3}</math></li> <li>• 0x03 - <math>1 \times 10^{-3}</math></li> <li>• 0x04 - <math>1 \times 10^{-4}</math></li> <li>• 0x05 - <math>1 \times 10^{-5}</math></li> <li>• 0x06 - <math>1 \times 10^{-6}</math></li> <li>• 0x07 - <math>1 \times 10^{-1}</math></li> </ul>
<i>resBerRatio</i>	<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>	- delivery of erroneous SDUs <ul style="list-style-type: none"> <li>Indicates whether SDUs detected as erroneous shall be delivered or not.</li> <li>0x00 - Subscribe</li> <li>0x01 - <math>5 \cdot 10^{-2}</math></li> <li>0x02 - <math>1 \cdot 10^{-2}</math></li> <li>0x03 - <math>5 \cdot 10^{-3}</math></li> <li>0x04 - <math>4 \cdot 10^{-3}</math></li> <li>0x05 - <math>1 \cdot 10^{-3}</math></li> <li>0x06 - <math>1 \cdot 10^{-4}</math></li> <li>0x07 - <math>1 \cdot 10^{-5}</math></li> <li>0x08 - <math>1 \cdot 10^{-6}</math></li> <li>0x09 - <math>1 \cdot 10^{-8}</math></li> </ul>
<i>transferDelay</i>	- Transfer delay (ms) <ul style="list-style-type: none"> <li>Indicates the targeted time between a request to transfer an SDU at one SAP to its delivery at the other SAP in milliseconds.</li> </ul>
<i>trafficPriority</i>	- Transfer handling priority <ul style="list-style-type: none"> <li>Specifies the relative importance for handling of SDUs that belong to the UMTS bearer, compared to the SDUs of other bearers.</li> </ul>

## 8.1206.2 Field Documentation

- 8.1206.2.1 `uint8_t wds_UMTSMinQoS::deliveryErrSDU`
- 8.1206.2.2 `uint32_t wds_UMTSMinQoS::grntDownlinkBitrate`
- 8.1206.2.3 `uint32_t wds_UMTSMinQoS::grntUplinkBitrate`
- 8.1206.2.4 `uint32_t wds_UMTSMinQoS::maxDownlinkBitrate`
- 8.1206.2.5 `uint32_t wds_UMTSMinQoS::maxSDUSize`
- 8.1206.2.6 `uint32_t wds_UMTSMinQoS::maxUplinkBitrate`
- 8.1206.2.7 `uint8_t wds_UMTSMinQoS::qosDeliveryOrder`
- 8.1206.2.8 `uint8_t wds_UMTSMinQoS::resBerRatio`
- 8.1206.2.9 `uint8_t wds_UMTSMinQoS::sduErrorRatio`
- 8.1206.2.10 `uint8_t wds_UMTSMinQoS::trafficClass`
- 8.1206.2.11 `uint32_t wds_UMTSMinQoS::trafficPriority`
- 8.1206.2.12 `uint32_t wds_UMTSMinQoS::transferDelay`

## 8.1207 wdsDhcpv4HwConfig Struct Reference

## Data Fields

- uint8\_t [hwType](#)
- uint8\_t [chaddrLen](#)
- uint8\_t [chaddr](#) [16]

### 8.1207.1 Detailed Description

Structure contain DHCP V4 Hardware Configuration.

#### 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

### 8.1207.2 Field Documentation

8.1207.2.1 uint8\_t wdsDhcpv4HwConfig::chaddr[16]

8.1207.2.2 uint8\_t wdsDhcpv4HwConfig::chaddrLen

8.1207.2.3 uint8\_t wdsDhcpv4HwConfig::hwType

## 8.1208 wdsDhcpv4Option Struct Reference

## Data Fields

- uint8\_t [optCode](#)
- uint8\_t [optValLen](#)
- uint8\_t [optVal](#) [255]

### 8.1208.1 Detailed Description

This structure contains DHCP V4 Option information.

#### 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

### 8.1208.2 Field Documentation



8.1208.2.1 `uint8_t wdsDhcpv4Option::optCode`

8.1208.2.2 `uint8_t wdsDhcpv4Option::optVal[255]`

8.1208.2.3 `uint8_t wdsDhcpv4Option::optValLen`

## 8.1209 wdsDhcpv4OptionList Struct Reference

### Data Fields

- `uint8_t numOpt`
- `wdsDhcpv4Option * pOptList`

### 8.1209.1 Detailed Description

Structure contain DHCP V4 Option List

#### 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 <ul style="list-style-type: none"> <li>• See <a href="#">wdsDhcpv4Option</a></li> </ul>

### 8.1209.2 Field Documentation

8.1209.2.1 `uint8_t wdsDhcpv4OptionList::numOpt`

8.1209.2.2 `wdsDhcpv4Option* wdsDhcpv4OptionList::pOptList`

## 8.1210 wdsDhcpv4ProfileId Struct Reference

### Data Fields

- `uint8_t profileType`
- `uint8_t profileId`

### 8.1210.1 Detailed Description

This structure contains DHCP v4 ProfileID Information.

#### 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>

## 8.1210.2 Field Documentation

8.1210.2.1 `uint8_t wdsDhcpv4ProfileId::profileId`

8.1210.2.2 `uint8_t wdsDhcpv4ProfileId::profileType`

## 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 audio.h File Reference

#### Data Structures

- struct [pack\\_audio\\_SLQSGetAudioProfile\\_t](#)
- struct [unpack\\_audio\\_SLQSGetAudioProfile\\_t](#)
- struct [pack\\_audio\\_SLQSSetAudioProfile\\_t](#)
- struct [pack\\_audio\\_SLQSGetAudioPathConfig\\_t](#)
- struct [audio\\_TXPCMIIRFitr](#)
- struct [audio\\_RXPCMIIRFitr](#)
- struct [audio\\_RXAGCList](#)
- struct [audio\\_RXAVCList](#)
- struct [audio\\_TXAGCList](#)
- struct [unpack\\_audio\\_SLQSGetAudioPathConfig\\_t](#)
- struct [pack\\_audio\\_SLQSSetAudioPathConfig\\_t](#)
- struct [pack\\_audio\\_SLQSGetAudioVolTLBConfig\\_t](#)
- struct [unpack\\_audio\\_SLQSGetAudioVolTLBConfig\\_t](#)
- struct [pack\\_audio\\_SLQSSetAudioVolTLBConfig\\_t](#)
- struct [unpack\\_audio\\_SLQSSetAudioVolTLBConfig\\_t](#)

## Functions

- int [pack\\_audio\\_SLQSGetAudioProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSGetAudioProfile\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSGetAudioProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_audio\\_SLQSGetAudioProfile\\_t](#) \*pOutput)
- int [pack\\_audio\\_SLQSSetAudioProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSSetAudioProfile\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSSetAudioProfile](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_audio\\_SLQSGetAudioPathConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSGetAudioPathConfig\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSGetAudioPathConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_audio\\_SLQSGetAudioPathConfig\\_t](#) \*pOutput)
- int [pack\\_audio\\_SLQSSetAudioPathConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSSetAudioPathConfig\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSSetAudioPathConfig](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [pack\\_audio\\_SLQSGetAudioVoTLBConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSGetAudioVoTLBConfig\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSGetAudioVoTLBConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_audio\\_SLQSGetAudioVoTLBConfig\\_t](#) \*pOutput)
- int [pack\\_audio\\_SLQSSetAudioVoTLBConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_audio\\_SLQSSetAudioVoTLBConfig\\_t](#) \*pReqParam)
- int [unpack\\_audio\\_SLQSSetAudioVoTLBConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_audio\\_SLQSSetAudioVoTLBConfig\\_t](#) \*pOutput)

### 9.2.1 Function Documentation

9.2.1.1 int [pack\\_audio\\_SLQSGetAudioPathConfig](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_audio\\_SLQSGetAudioPathConfig\\_t](#) \* *pReqParam* )

Gets the audio path configuration parameters 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>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

#### Note

Device Supported: SL9090

9.2.1.2 int [pack\\_audio\\_SLQSGetAudioProfile](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_audio\\_SLQSGetAudioProfile\\_t](#) \* *pReqParam* )

Gets get the profile content of the requested audio generator. 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>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

## Note

Device Supported: SL9090

**9.2.1.3** `int pack_audio_SLQSGetAudioVolTLBConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_audio_SLQSGetAudioVolTLBConfig_t * pReqParam )`

Gets the audio VolTLB configuration parameters 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>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

## Note

Device Supported: SL9090

**9.2.1.4** `int pack_audio_SLQSSetAudioPathConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_audio_SLQSSetAudioPathConfig_t * pReqParam )`

sets the audio path configuration parameters 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>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

**Note**

Device Supported: SL9090

**9.2.1.5** int pack\_audio\_SLQSSetAudioProfile ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_audio\_SLQSSetAudioProfile\_t \* pReqParam )

sets an audio profile pack.

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReqParam	request parameters

**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

**9.2.1.6** int pack\_audio\_SLQSSetAudioVolTLBConfig ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_audio\_SLQSSetAudioVolTLBConfig\_t \* pReqParam )

Sets the audio VolTLB configuration parameters pack.

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReqParam	request parameters

**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

**9.2.1.7** `int unpack_audio_SLQSGetAudioPathConfig ( uint8_t * pResp, uint16_t respLen, unpack_audio_SLQSGetAudioPathConfig_t * pOutput )`

Gets the audio path configuration parameters 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

**9.2.1.8** `int unpack_audio_SLQSGetAudioProfile ( uint8_t * pResp, uint16_t respLen, unpack_audio_SLQSGetAudioProfile_t * pOutput )`

Gets get the profile content of the requested audio generator 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

**9.2.1.9** `int unpack_audio_SLQSGetAudioVoTLBConfig ( uint8_t * pResp, uint16_t respLen, unpack_audio_SLQSGetAudioVoTLBConfig_t * pOutput )`

Gets the audio VoTLB configuration parameters 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

#### 9.2.1.10 int unpack\_audio\_SLQSSetAudioPathConfig ( uint8\_t \* *pResp*, uint16\_t *respLen* )

sets the audio path configuration parameters 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

#### 9.2.1.11 int unpack\_audio\_SLQSSetAudioProfile ( uint8\_t \* *pResp*, uint16\_t *respLen* )

sets an audio 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

#### 9.2.1.12 int unpack\_audio\_SLQSSetAudioVoTLBConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_audio\_SLQSSetAudioVoTLBConfig\_t \* *pOutput* )

Sets the audio VoTLB configuration parameters 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

## 9.3 cat.h File Reference

**Data Structures**

- struct [pack\\_cat\\_SetCATEventCallback\\_t](#)
- struct [unpack\\_cat\\_SetCATEventCallback\\_t](#)
- struct [cat\\_EventIDDataTlv](#)
- struct [cat\\_AIPhIdIdentifierTlv](#)
- struct [cat\\_EventListTlv](#)
- struct [cat\\_RefreshTlv](#)
- struct [cat\\_EndProactiveSessionTlv](#)
- union [cat\\_currentCatEvent](#)
- struct [cat\\_commonEventTlv](#)
- struct [unpack\\_cat\\_SetCatEventCallback\\_ind\\_t](#)
- struct [pack\\_cat\\_CATSendEnvelopeCommand\\_t](#)
- struct [pack\\_cat\\_CATSendTerminalResponse\\_t](#)

**Macros**

- #define [CAT\\_EVENT\\_DATA\\_MAX\\_LENGTH](#) 255
- #define [CAN\\_COMMON\\_EVENT\\_TLV\\_NUMBER](#) 11

**Typedefs**

- typedef [unpack\\_result\\_t](#) [unpack\\_cat\\_CATSendEnvelopeCommand\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_cat\\_CATSendTerminalResponse\\_t](#)

**Functions**

- int [pack\\_cat\\_SetCATEventCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_cat\\_SetCATEventCallback\\_t](#) \*reqArg)
- int [unpack\\_cat\\_SetCATEventCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_cat\\_SetCATEventCallback\\_t](#) \*pOutput)
- int [unpack\\_cat\\_SetCatEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_cat\\_SetCatEventCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_cat\\_CATSendEnvelopeCommand](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_cat\\_CATSendEnvelopeCommand\\_t](#) \*reqArg)
- int [unpack\\_cat\\_CATSendEnvelopeCommand](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_cat\\_CATSendEnvelopeCommand\\_t](#) \*pOutput)
- int [pack\\_cat\\_CATSendTerminalResponse](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_cat\\_CATSendTerminalResponse\\_t](#) \*reqArg)
- int [unpack\\_cat\\_CATSendTerminalResponse](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_cat\\_CATSendTerminalResponse\\_t](#) \*pOutput)

### 9.3.1 Macro Definition Documentation

9.3.1.1 `#define CAN_COMMON_EVENT_TLV_NUMBER 11`

9.3.1.2 `#define CAT_EVENT_DATA_MAX_LENGTH 255`

### 9.3.2 Typedef Documentation

9.3.2.1 `typedef unpack_result_t unpack_cat_CATSendEnvelopeCommand_t`

9.3.2.2 `typedef unpack_result_t unpack_cat_CATSendTerminalResponse_t`

### 9.3.3 Function Documentation

9.3.3.1 `int pack_cat_CATSendEnvelopeCommand ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_cat_CATSendEnvelopeCommand_t * reqArg )`

Sends the envelope command to the device 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

9.3.3.2 `int pack_cat_CATSendTerminalResponse ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_cat_CATSendTerminalResponse_t * reqArg )`

Sends the terminal response to the device 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

9.3.3.3 `int pack_cat_SetCATEventCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_cat_SetCATEventCallback_t * reqArg )`

Enables/disables the CAT 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
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

9.3.3.4 `int unpack_cat_CATSendEnvelopeCommand ( uint8_t * pResp, uint16_t respLen, unpack_cat_CATSendEnvelopeCommand_t * pOutput )`

Sends the envelope command to the device 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

9.3.3.5 `int unpack_cat_CATSendTerminalResponse ( uint8_t * pResp, uint16_t respLen, unpack_cat_CATSendTerminalResponse_t * pOutput )`

Sends the terminal response to the device 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

**9.3.3.6** `int unpack_cat_SetCATEventCallback ( uint8_t * pResp, uint16_t respLen, unpack_cat_SetCATEventCallback_t * pOutput )`

Enables/disables the CAT event callback 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

**9.3.3.7** `int unpack_cat_SetCatEventCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_cat_SetCatEventCallback_ind_t * pOutput )`

CAT event indication 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

## 9.4 common.h File Reference

**Data Structures**

- struct [pack\\_qmi\\_t](#)
- struct [unpack\\_qmi\\_t](#)
- struct [unpack\\_result\\_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)
- #define [SDK\\_VALIDATE\\_INPUT\\_PACK\\_PARAM\\_AND\\_FILL\\_XID](#)(pCtx, pBuf, pLen)
- #define [SDK\\_VALIDATE\\_INPUT\\_UNPACK\\_PARAM](#)(pResp, respLen, pOutput)
- #define [SWI\\_INIT\\_UNPACK\\_RESULT\\_VALUE](#) {0, [SWI\\_UINT256\\_INT\\_VALUE](#)}

## 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,  
[ePDS](#) =0x06,  
[eVOICE](#) =0x09,  
[eCAT](#) =0x0A,  
[eUIM](#) =0x0B,  
[eLOC](#) =0x10,  
[eSAR](#) =0x11,  
[eIMS](#) =0x12,  
[eTMD](#) =0x18,  
[eIMSA](#) =0x21,  
[eRMS](#) =225,  
[eSWIOMA](#) =240,  
[eAUDIO](#) =241,  
[eSWIM2MCMD](#) =243,  
[eSWIM2MCMD\\_AVC2](#) =0xfd,  
[eSWIOMAEXT](#) =244,  
[eSWIAUDIO](#) = 245,  
[eSWILOC](#) =246,  
[eSWIDMS](#) =254 }
- enum [msgtype](#) {  
[eREQ](#) =0,  
[eRSP](#) =2,  
[eIND](#) =4 }

## Functions

- uint16\_t [helper\\_get\\_xid](#) (uint8\_t \*qmi\_resp)
- uint16\_t [helper\\_get\\_error\\_code](#) (uint8\_t \*qmi\_resp)
- char \* [helper\\_get\\_error\\_reason](#) (uint16\_t retVal)
- const char \* [helper\\_get\\_resp\\_ctx](#) (uint8\_t svc, uint8\_t \*pbuf, uint32\_t len, [unpack\\_qmi\\_t](#) \*pCtx)
- const char \* [helper\\_get\\_req\\_str](#) (uint8\_t svc, uint8\_t \*req, uint32\_t len)
- unsigned [unpack\\_result\\_code\\_only](#) (uint8\_t \*pMdmResp)
- int [helper\\_set\\_log\\_func](#) (logger \*func)
- void [liteqmi\\_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 \* [liteqmi\\_GetVersion](#) ()
- void [liteqmi\\_helper\\_decode7bitAsciiEncString](#) (uint8\_t \*encoded\_str, uint8\_t encoded\_len, uint8\_t \*decoded\_str)
- int [helper\\_isBootLoader\\_DebugEnabled](#) (const char \*szPath, const char \*pQsn)

## Variables

- [logger](#) \* [glog](#)
- uint8\_t [gloglvl](#)

### 9.4.1 Macro Definition Documentation

9.4.1.1 **#define** DEault\_LOC\_TIMEOUT\_IN\_SEC 2

9.4.1.2 **#define** MINREQBKLEN (2048)

9.4.1.3 **#define** MSGID\_AND\_LEN (4)

9.4.1.4 **#define** MSGID\_DONT\_CARE (0xffff)

9.4.1.5 **#define** SDK\_VALIDATE\_INPUT\_PACK\_PARAM( *pCtx*, *pBuf*, *pLen* )

**Value:**

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
{
    liteqmi_log(eLOG_DEBUG, "[ pack] %s parameter NULL\n", __func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
}
```

9.4.1.6 **#define** SDK\_VALIDATE\_INPUT\_PACK\_PARAM\_AND\_FILL\_XID( *pCtx*, *pBuf*, *pLen* )

**Value:**

```
if ((pCtx == NULL) || (pBuf == NULL) || (pLen == NULL) ) \
{
    liteqmi_log(eLOG_DEBUG, "[ pack] %s parameter NULL\n", __func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
} \
if (pCtx->xid == 0) \
    return eQCWWAN_ERR_INVALID_XID; \
pBuf[0] = eREQ; \
pBuf[1] = pCtx->xid & 0xff; \
pBuf[2] = pCtx->xid >> 8;
```

9.4.1.7 #define SDK\_VALIDATE\_INPUT\_UNPACK\_PARAM( *pResp*, *respLen*, *pOutput* )**Value:**

```

if ((pResp == NULL) || (respLen == 0) || (pOutput == NULL) ) \
{
    liteqmi_log(eLOG_DEBUG, "[ unpack] %s parameter NULL or invalid\n",
__func__); \
    return eQCWWAN_ERR_INVALID_ARG; \
}

```

## 9.4.1.8 #define SDU\_HDR\_LEN (3)

## 9.4.1.9 #define SWI\_INIT\_UNPACK\_RESULT\_VALUE {0, SWI\_UINT256\_INT\_VALUE}

9.4.1.10 #define UNUSEDPARAM( *x* )(void)x

## 9.4.2 Enumeration Type Documentation

## 9.4.2.1 enum eLOG\_LEVEL

log levels

Enumerator

***eLOG\_INFO***  
***eLOG\_DEBUG***  
***eLOG\_WARN***  
***eLOG\_FATAL***

## 9.4.2.2 enum eQMI\_SVC

qmi service

Enumerator

***eCTL***  
***eWDS***  
***eDMS***  
***eNAS***  
***eQOS***  
***eSMS***  
***ePDS***  
***eVOICE***  
***eCAT***  
***eUIM***  
***eLOC***  
***eSAR***  
***eIMS***  
***eTMD***  
***eIMSA***  
***eRMS***

***eSWIOMA***  
***eAUDIO***  
***eSWIM2MCMD***  
***eSWIM2MCMD\_AVC2***  
***eSWIOMAEXT***  
***eSWIAUDIO***  
***eSWILOC***  
***eSWIDMS***

#### 9.4.2.3 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***

#### 9.4.2.4 enum msgtype

qmi message type

Enumerator

***eREQ***  
***eRSP***  
***eIND***

### 9.4.3 Function Documentation

9.4.3.1 void fill\_pack\_ctx ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, uint8\_t svc, int timeout )

9.4.3.2 void fill\_sdu\_hdr ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf )

9.4.3.3 char\* get\_version ( )

Returns

version string



9.4.3.4 `uint16_t helper_get_error_code ( uint8_t * qmi_resp )`

9.4.3.5 `char* helper_get_error_reason ( uint16_t retVal )`

9.4.3.6 `const char* helper_get_req_str ( uint8_t svc, uint8_t * req, uint32_t len )`

extract msgid string from modem req

#### Parameters

in	<i>svc</i>	qmi service
in	<i>req</i>	qmi request
in	<i>len</i>	request length

#### Returns

qmi message string

9.4.3.7 `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	<i>svc</i>	qmi service
in	<i>pbuf</i>	qmi response/indication
in	<i>len</i>	response/indication length
out	<i>pCtx</i>	unpacked context

#### Returns

qmi message string

9.4.3.8 `uint16_t helper_get_xid ( uint8_t * qmi_resp )`

9.4.3.9 `int helper_isBootLoader_DebugEnabled ( const char * szPath, const char * pQsn )`

#### Parameters

in	<i>szPath</i>	the full path of the provided bootloader or firmware file
in	<i>pQsn</i>	the QSN of a specific module user wants to check, it is a 4 bytes hex value without the prefix 0X. It can be retrieved by AT command AT!ENTERCND="<password>", AT!SECBOOTCFG?, the first parameter of the response of AT!SECBOOTCFG?.

#### Returns

values listed below

- -1 bootloader is not debug enabled
- 0 unable to retrieve the bootloader debug info
- 1 bootloader is debug enabled for the provided QSN
- 4 either *szPath* or *pQsn* is invalid (NULL)
- 18 invalid file

9.4.3.10 `int helper_set_log_func ( logger * func )`

set log function

9.4.3.11 `int helper_set_log_lvl ( uint8_t lvl )`

set log level

9.4.3.12 `char* liteqmi_GetVersion ( )`

Returns

version string

9.4.3.13 `void liteqmi_helper_decode7bitAsciiEncString ( uint8_t * encoded_str, uint8_t encoded_len, uint8_t * decoded_str )`

decode 7bit encoded string

Parameters

in	<i>encoded_str</i>	7 bit encoded ASCII
in	<i>encoded_len</i>	encoded length
out	<i>decoded_str</i>	decoded ASCII string

9.4.3.14 `void liteqmi_log ( uint8_t lvl, const char * fmt, ... )`

9.4.3.15 `unsigned unpack_result_code_only ( uint8_t * pMdmResp )`

common handler for unpacking response with TLV type 0x02 only

## 9.4.4 Variable Documentation

9.4.4.1 `logger* glog`

9.4.4.2 `uint8_t gloglvl`

## 9.5 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 [dms\\_LteBandsSupport](#)
- struct [unpack\\_dms\\_SLQSGetBandCapabilityExt\\_t](#)
- struct [unpack\\_dms\\_GetDeviceCapabilities\\_t](#)
- struct [dms\\_devCaps](#)
- struct [dms\\_devSubsCfgList](#)

- struct [dms\\_devMultiSimCaps](#)
- struct [dms\\_devMultiSimVoiceDataCaps](#)
- struct [dms\\_devCurSubsCaps](#)
- struct [dms\\_devSubsVoiceDataList](#)
- struct [dms\\_devSubsVoiceDataCaps](#)
- struct [dms\\_devSubsFeatureModeCaps](#)
- struct [dms\\_devMaxSubsCaps](#)
- struct [dms\\_devSubsList](#)
- struct [dms\\_devMaxCfgListCaps](#)
- struct [unpack\\_dms\\_GetDeviceCapabilitiesV2\\_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\\_GetNetworkTimeV2\\_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](#)
- struct [unpack\\_dms\\_GetManufacturer\\_t](#)
- struct [unpack\\_dms\\_GetOfflineReason\\_t](#)
- struct [pack\\_dms\\_SetActivationStatusCallback\\_t](#)
- struct [unpack\\_dms\\_SetActivationStatusCallback\\_t](#)
- struct [pack\\_dms\\_UIMSetPINProtection\\_t](#)
- struct [unpack\\_dms\\_UIMSetPINProtection\\_t](#)
- struct [pack\\_dms\\_UIMUnblockPIN\\_t](#)
- struct [pack\\_dms\\_UIMVerifyPIN\\_t](#)
- struct [pack\\_dms\\_UIMChangePIN\\_t](#)
- struct [pack\\_dms\\_UIMGetControlKeyStatus\\_t](#)
- struct [unpack\\_dms\\_UIMGetControlKeyStatus\\_t](#)
- struct [unpack\\_dms\\_UIMGetPINStatus\\_t](#)
- struct [pack\\_dms\\_UIMSetControlKeyProtection\\_t](#)
- struct [unpack\\_dms\\_UIMSetControlKeyProtection\\_t](#)
- struct [pack\\_dms\\_UIMUnblockControlKey\\_t](#)
- struct [unpack\\_dms\\_UIMUnblockControlKey\\_t](#)
- struct [pack\\_dms\\_ResetToFactoryDefaults\\_t](#)
- struct [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#)
- struct [pack\\_dms\\_ActivateAutomatic\\_t](#)
- struct [eriDataparams](#)
- struct [unpack\\_dms\\_SLQSGetERIFile\\_t](#)
- struct [unpack\\_dms\\_SLQSUIMGetState\\_t](#)
- struct [pack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#)
- struct [crashInformation](#)
- struct [crashInfoParams](#)
- struct [unpack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetHostDevInfo\\_t](#)
- struct [pack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetOSInfo\\_t](#)
- struct [pack\\_dms\\_SLQSSwiSetOSInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiSetOSInfo\\_t](#)
- struct [unpack\\_dms\\_SLQSSwiGetSerialNoExt\\_t](#)
- struct [unpack\\_dms\\_SLQSDmsSwiGetPCInfo\\_t](#)
- struct [dms\\_PSMEnableStateTlv](#)
- struct [dms\\_PSMDurationThresholdTlv](#)
- struct [dms\\_PSMDurationDueToOOSTlv](#)
- struct [dms\\_PSMRandomizationWindowTlv](#)
- struct [dms\\_PSMActiveTimerTlv](#)
- struct [dms\\_PSMPeriodicUpdateTimerTlv](#)
- struct [dms\\_PSMEarlyWakeupTimeTlv](#)
- struct [unpack\\_dms\\_SLQSGetPowerSaveModeConfig\\_t](#)
- struct [pack\\_dms\\_SLQSSetPowerSaveModeConfig\\_t](#)
- struct [unpack\\_dms\\_SLQSSetPowerSaveModeConfig\\_t](#)
- struct [pack\\_dms\\_SetIndicationRegister\\_t](#)
- struct [unpack\\_dms\\_SetIndicationRegister\\_t](#)

- struct [dms\\_PSMEnableStateIndTlv](#)
- struct [dms\\_PSMActiveTimerIndTlv](#)
- struct [dms\\_PSMPeriodicUpdateTimerIndTlv](#)
- struct [unpack\\_dms\\_PSMCfgChange\\_ind\\_t](#)
- struct [pack\\_dms\\_SwiSetEventReport\\_t](#)
- struct [unpack\\_dms\\_SwiSetEventReport\\_t](#)
- struct [dms\\_TemperatureTlv](#)
- struct [dms\\_VoltageTlv](#)
- struct [dms\\_UimStatusTlv](#)
- struct [unpack\\_dms\\_SwiEventReportCallBack\\_ind\\_t](#)
- struct [pack\\_dms\\_SwiUimSelect\\_t](#)
- struct [unpack\\_dms\\_SwiUimSelect\\_t](#)
- struct [dms\\_UimAutoSwitchActSlotTlv](#)
- struct [unpack\\_dms\\_SLQSDmsSwiGetUimSelection\\_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\\_VALID\\_FSN\\_LEN](#) 14
- [#define DMS\\_MAX\\_RADIO\\_IFCS\\_SIZE](#) 255
- [#define DMS\\_MAX\\_SUBS\\_CFG\\_LIST\\_SIZE](#) 32
- [#define DMS\\_MAX\\_SUBS\\_LIST\\_SIZE](#) 32
- [#define DMS\\_MAX\\_SUPPORTED\\_LTE\\_BANDS](#) 255
- [#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 SPC\\_SIZE](#) 6
- [#define CK\\_MAX\\_SIZE](#) 8
- [#define ACT\\_CODE\\_MAX\\_SIZE](#) 81
- [#define ERI\\_DATA\\_MAX\\_SIZE](#) 1024
- [#define MEID\\_MAX\\_SIZE](#) 8
- [#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`
- `#define DMS_SET_REG_IND_ENABLE 1`
- `#define DMS_SET_REG_IND_DISABLE 0`
- `#define DMS_SET_REG_IND_NO_CHANGE 2`

## 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 unpack_dms_SLQSGetBandCapabilityExt (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSGetBandCapabilityExt_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_GetDeviceCapabilitiesV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetDeviceCapabilitiesV2 (uint8_t *pResp, uint16_t respLen, unpack_dms_GetDeviceCapabilitiesV2_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_GetNetworkTimeV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, void *reqArg)`
- `int unpack_dms_GetNetworkTimeV2 (uint8_t *pResp, uint16_t respLen, unpack_dms_GetNetworkTimeV2_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_SLQSDmsSwtGetResetInfo (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_dms_SLQSDmsSwtGetResetInfo (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwtGetResetInfo_t *pOutput)`
- `int unpack_dms_SLQSDmsSwtGetResetInfo_Ind (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwtGetResetInfo_Ind_t *pOutput)`
- `int pack_dms_SLQSDmsSwtIndicationRegister (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SLQSDmsSwtIndicationRegister_t *reqArg)`
- `int unpack_dms_SLQSDmsSwtIndicationRegister (uint8_t *pResp, uint16_t respLen, unpack_dms_SLQSDmsSwtIndicationRegister_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)`
- `int pack_dms_GetManufacturer (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_dms_GetManufacturer (uint8_t *pResp, uint16_t respLen, unpack_dms_GetManufacturer_t *pOutput)`
- `int pack_dms_GetOfflineReason (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_dms_GetOfflineReason (uint8_t *pResp, uint16_t respLen, unpack_dms_GetOfflineReason_t *pOutput)`
- `int pack_dms_SetActivationStatusCallback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_SetActivationStatusCallback_t *reqArg)`
- `int unpack_dms_SetActivationStatusCallback (uint8_t *pResp, uint16_t respLen, unpack_dms_SetActivationStatusCallback_t *pOutput)`
- `int pack_dms_UIMSetPINProtection (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMSetPINProtection_t *pReq)`
- `int unpack_dms_UIMSetPINProtection (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)`
- `int pack_dms_UIMUnblockPIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMUnblockPIN_t *pReq)`
- `int unpack_dms_UIMUnblockPIN (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)`
- `int pack_dms_UIMVerifyPIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMVerifyPIN_t *pReq)`
- `int unpack_dms_UIMVerifyPIN (uint8_t *pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t *pOutput)`
- `int pack_dms_UIMChangePIN (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_dms_UIMChangePIN_t *pReq)`



- int [unpack\\_dms\\_UIMChangePIN](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetPINProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMGetControlKeyStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMGetControlKeyStatus\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMGetControlKeyStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMGetControlKeyStatus\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMGetPINStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_UIMGetPINStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMGetPINStatus\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMSetControlKeyProtection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMSetControlKeyProtection\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMSetControlKeyProtection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMSetControlKeyProtection\\_t](#) \*pOutput)
- int [pack\\_dms\\_UIMUnblockControlKey](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_UIMUnblockControlKey\\_t](#) \*pReq)
- int [unpack\\_dms\\_UIMUnblockControlKey](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_UIMUnblockControlKey\\_t](#) \*pOutput)
- int [pack\\_dms\\_ResetToFactoryDefaults](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pReq)
- int [unpack\\_dms\\_ResetToFactoryDefaults](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_ValidateSPC](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pReq)
- int [unpack\\_dms\\_ValidateSPC](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_ActivateAutomatic](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_ActivateAutomatic\\_t](#) \*pReq)
- int [unpack\\_dms\\_ActivateAutomatic](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_ResetToFactoryDefaults\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSGetERIFile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSGetERIFile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSGetERIFile\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSUIGetState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSUIGetState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSUIGetState\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetCrashInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiGetCrashInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetCrashInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetHostDevInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetHostDevInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetHostDevInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiSetHostDevInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiSetHostDevInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiSetHostDevInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetOSInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetOSInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetOSInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiSetOSInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSwiSetOSInfo\\_t](#) \*pReq)
- int [unpack\\_dms\\_SLQSSwiSetOSInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiSetOSInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSwiGetSerialNoExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_dms\\_SLQSSwiGetSerialNoExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSwiGetSerialNoExt\\_t](#) \*pOutput)

- int [pack\\_dms\\_SLQSDmsSwiGetPCInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_SLQSDmsSwiGetPCInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSDmsSwiGetPCInfo\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSGetPowerSaveModeConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_SLQSGetPowerSaveModeConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSGetPowerSaveModeConfig\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSSetPowerSaveModeConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SLQSSetPowerSaveModeConfig\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SLQSSetPowerSaveModeConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSSetPowerSaveModeConfig\\_t](#) \*pOutput)
- int [pack\\_dms\\_SetIndicationRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SetIndicationRegister\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SetIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SetIndicationRegister\\_t](#) \*pOutput)
- int [unpack\\_dms\\_PSMCfgChange\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_PSMCfgChange\\_ind\\_t](#) \*pOutput)
- int [pack\\_dms\\_SwiSetEventReport](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SwiSetEventReport\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SwiSetEventReport](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SwiSetEventReport\\_t](#) \*pOutput)
- int [unpack\\_dms\\_SwiEventReportCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SwiEventReportCallBack\\_ind\\_t](#) \*pOutput)
- int [pack\\_dms\\_SwiUimSelect](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_dms\\_SwiUimSelect\\_t](#) \*reqArg)
- int [unpack\\_dms\\_SwiUimSelect](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SwiUimSelect\\_t](#) \*pOutput)
- int [pack\\_dms\\_SLQSDmsSwiGetUimSelection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_dms\\_SLQSDmsSwiGetUimSelection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_dms\\_SLQSDmsSwiGetUimSelection\\_t](#) \*pOutput)

## 9.5.1 Macro Definition Documentation

9.5.1.1 `#define ACT_CODE_MAX_SIZE 81`

9.5.1.2 `#define CK_MAX_SIZE 8`

9.5.1.3 `#define DMS_IMGDETAILS_LEN 16`

9.5.1.4 `#define DMS_MAX_CUST_ID_LEN 64`

9.5.1.5 `#define DMS_MAX_CUST_VALUE_LEN 8`

9.5.1.6 `#define DMS_MAX_FWUPDATE_LOG_STR_SZ 255`

9.5.1.7 `#define DMS_MAX_FWUPDATE_REF_STR_SZ 15`

9.5.1.8 `#define DMS_MAX_RADIO_IFCS_SIZE 255`

9.5.1.9 `#define DMS_MAX_SUBS_CFG_LIST_SIZE 32`

9.5.1.10 `#define DMS_MAX_SUBS_LIST_SIZE 32`

9.5.1.11 `#define DMS_MAX_SUPPORTED_LTE_BANDS 255`

9.5.1.12 `#define DMS_PM_FACTORY 0x02 /* Factory Test Mode */`

9.5.1.13 `#define DMS_PM_LOW 0x01 /* Low Power */`

9.5.1.14 `#define DMS_PM_OFFLINE 0x03 /* Offline */`

9.5.1.15 `#define DMS_PM_ONLINE 0x00 /* Online */`

9.5.1.16 `#define DMS_PM_PERSISTENT_LOW 0x06 /* Persistent Low Power */`

9.5.1.17 `#define DMS_PM_RESET 0x04 /* Reset */`

9.5.1.18 `#define DMS_PM_SHUT_DOWN 0x05 /* Shut Down */`

9.5.1.19 `#define DMS_SET_REG_IND_DISABLE 0`

9.5.1.20 `#define DMS_SET_REG_IND_ENABLE 1`

9.5.1.21 `#define DMS_SET_REG_IND_NO_CHANGE 2`

9.5.1.22 `#define DMS_SET_REPORT_DISABLE 0`

9.5.1.23 `#define DMS_SET_REPORT_ENABLE 1`

9.5.1.24 `#define DMS_SLQSFWINFO_APPVERSION_SZ 85`

9.5.1.25 `#define DMS_SLQSFWINFO_BOOTVERSION_SZ 85`

9.5.1.26 `#define DMS_SLQSFWINFO_CARRIER_SZ 20`

9.5.1.27 `#define DMS_SLQSFWINFO_CUR_CARR_NAME 17`

9.5.1.28 `#define DMS_SLQSFWINFO_CUR_CARR_REV 13`

9.5.1.29 `#define DMS_SLQSFWINFO_MODELID_SZ 20`

9.5.1.30 `#define DMS_SLQSFWINFO_PACKAGEID_SZ 85`

9.5.1.31 `#define DMS_SLQSFWINFO_PRIVERSION_SZ 16`

9.5.1.32 `#define DMS_SLQSFWINFO_SKU_SZ 15`

9.5.1.33 `#define DMS_SWI_SET_IND_DISABLE 0`

9.5.1.34 `#define DMS_SWI_SET_IND_ENABLE 1`

9.5.1.35 `#define DMS_UINT8_MAX_STRING_SZ 255`

9.5.1.36 `#define DMS_VALID_FSN_LEN 14`

9.5.1.37 `#define ERI_DATA_MAX_SIZE 1024`

9.5.1.38 `#define MAX_BUILD_ID_LEN 255`

9.5.1.39 `#define MEID_MAX_SIZE 8`

9.5.1.40 `#define SLQS_MAX_DYING_GASP_CFG_SMS_CONTENT_LENGTH 160`

9.5.1.41 `#define SLQS_MAX_DYING_GASP_CFG_SMS_NUMBER_LENGTH 20`

9.5.1.42 `#define SPC_SIZE 6`

9.5.1.43 `#define UNIQUE_ID_LEN 16`

## 9.5.2 Function Documentation

9.5.2.1 `int pack_dms_ActivateAutomatic ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ActivateAutomatic_t * pReq )`

Requests the device to perform automatic service activation pack

### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>pReq</i>	requset parameter

### Returns

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

### See Also

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

9.5.2.2 `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

### Note

This is only supported for 3GPP2 devices.

9.5.2.3 `int pack_dms_GetBandCapability ( 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(NULL)

## Returns

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

## See Also

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

**9.5.2.4** `int pack_dms_GetCrashAction ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Crash Action 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(NULL)

## Returns

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

## See Also

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

**9.5.2.5** `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, 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(NULL)

## Returns

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

## See Also

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

**9.5.2.6** `int pack_dms_GetCustFeaturesV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_GetCustFeaturesV2_t * reqArg )`

Get Custom Feature pack This function queries the modem for a list of supported features. This function is for firmware version 2.0 and newer. Currently supported Customization features:

- GPIOARENABLE
- GPSSEL
- IMSWITCHHIDE
- IPV6ENABLE
- WAKEHOSTEN

#### Parameters

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

**9.5.2.7** `int pack_dms_GetDeviceCap ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Device Capabilities 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(NULL)

#### Returns

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

#### See Also

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

**9.5.2.8** `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(NULL)

## Returns

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

## See Also

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

**9.5.2.9** `int pack_dms_GetDeviceCapabilitiesV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get device capability pack v2

## 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(NULL)

## Returns

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

## See Also

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

**9.5.2.10** `int pack_dms_GetDeviceHardwareRev ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Hardware Revision of the device 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(NULL)

## Returns

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

## See Also

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

**9.5.2.11** `int pack_dms_GetDeviceMfr ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Device Manufacture Name 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.12** `int pack_dms_GetDeviceSerialNumbers ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Device Serial Number 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.13** `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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.14** `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(NULL)



## Returns

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

## See Also

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

**9.5.2.15** `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(NULL)

## Returns

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

## See Also

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

**9.5.2.16** `int pack_dms_GetFSN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get FSN pack. This API get the Factory Sequence Number of the device.

## 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(NULL)

## Returns

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

## See Also

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

**9.5.2.17** `int pack_dms_GetHardwareRevision ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get hardware revision of the device 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.18 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.19 int pack\_dms\_GetManufacturer ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )**

To get device manufacturer information 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

**9.5.2.20 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(NULL)

## Returns

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

## See Also

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

9.5.2.21 int pack\_dms\_GetNetworkTime ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Get Network Time pack Returns the current time of the device based on the value supported by the network.

## 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(NULL)

## Returns

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

## See Also

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

9.5.2.22 int pack\_dms\_GetNetworkTimeV2 ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Get Network Time pack Returns the current time of the device based on the value supported by the network.

## 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(NULL)

## Returns

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

## See Also

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

### 9.5.2.23 int pack\_dms\_GetOfflineReason ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

To get operating mode offline reason 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

### 9.5.2.24 int pack\_dms\_GetPower ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get operating mode of the device 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(NULL)

#### Returns

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

#### See Also

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

### 9.5.2.25 int pack\_dms\_GetPRLVersion ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, void \* reqArg )

Get PRL Versions pack. Return version of the active Preferred Roaming List (PRL) in use by the device.

#### 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(NULL)

#### Returns

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

#### See Also

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

9.5.2.26 `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(NULL)

#### Returns

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

#### See Also

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

9.5.2.27 `int pack_dms_GetUSBComp ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get USB Comp pack This API queries the modem's USB interface configuration and supported configuration parameters.

#### 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(NULL)

#### Returns

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

#### See Also

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

9.5.2.28 `int pack_dms_GetVoiceNumber ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Voice Number pack. Returns the voice number in use by the device.

#### 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(NULL)

#### Returns

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

## See Also

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

**9.5.2.29** `int pack_dms_ResetToFactoryDefaults ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ResetToFactoryDefaults_t * pReq )`

Resets to default factory settings of the device pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>pReq</i>	requeset parameter

## Returns

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

## See Also

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

**9.5.2.30** `int pack_dms_SetActivationStatusCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetActivationStatusCallback_t * reqArg )`

Set activation 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
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

**9.5.2.31** `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, 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

**9.5.2.32** `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, 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

**9.5.2.33** `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, 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

**9.5.2.34** `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, 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

**9.5.2.35** int pack\_dms\_SetFirmwarePreference ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Set Firmware Preference 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.36** int pack\_dms\_SetIndicationRegister ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_SetIndicationRegister\_t \* *reqArg* )

Set Indication register 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



**9.5.2.37** `int pack_dms_SetPower ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetPower_t * reqArg )`

Set Power pack Sets the operating mode of the device.

#### 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

**9.5.2.38** `int pack_dms_SetUSBComp ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SetUSBComp_t * reqArg )`

Set USB Comp pack 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

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

**9.5.2.39** `int pack_dms_SLQSDmsSwiGetPCInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get device power control status information 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.40** `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

**9.5.2.41** `int pack_dms_SLQSDmsSwiGetUimSelection ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get device UIM Selection information 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(NULL)

**Returns**

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

**See Also**

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

**9.5.2.42** `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
in	<i>reqArg</i>	request parameter

## 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

**9.5.2.43** `int pack_dms_SLQSGetBandCapability ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get band capability of the device 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(NULL)

## Returns

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

## See Also

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

**9.5.2.44** `int pack_dms_SLQSGetERIFile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To set UIM Get ERI file 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

**9.5.2.45** `int pack_dms_SLQSGetPowerSaveModeConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Power Save Mode (PSM) configuration parameter 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(NULL)

#### Returns

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

#### See Also

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

**9.5.2.46** `int pack_dms_SLQSSetPowerSaveModeConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSetPowerSaveModeConfig_t * reqArg )`

Sets the Power Save Mode (PSM) configuration parameters 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

**9.5.2.47** `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

**9.5.2.48** `int pack_dms_SLQSSwiGetCrashInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiGetCrashInfo_t * pReq )`

To Get the Crash Information from the device pack.

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<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

**9.5.2.49** `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

**9.5.2.50** `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

#### 9.5.2.51 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

#### 9.5.2.52 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

#### 9.5.2.53 int pack\_dms\_SLQSSwiGetHostDevInfo ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

To get host dev information 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

#### 9.5.2.54 int pack\_dms\_SLQSSwiGetOSInfo ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

To Get Host OS 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

#### 9.5.2.55 int pack\_dms\_SLQSSwiGetSerialNoExt ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

To get serial number extension 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

#### 9.5.2.56 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
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

**9.5.2.57** `int pack_dms_SLQSSwiSetHostDevInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetHostDevInfo_t * pReq )`

To set Host Dev 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
out	<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

**9.5.2.58** `int pack_dms_SLQSSwiSetOSInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_SLQSSwiSetOSInfo_t * pReq )`

To set Host OS 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
out	<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

**9.5.2.59** `int pack_dms_SLQSUIGetState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Returns the UIM state pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9-X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [unpack\\_uim\\_GetCardStatus/ unpack\\_uim\\_GetCardStatus](#) 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

## Returns

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

## See Also

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

**9.5.2.60** int pack\_dms\_SwiSetEventReport ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_dms\_SwiSetEventReport\_t \* *reqArg* )

Sets the DMS swi event report parameters 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

**9.5.2.61** int pack\_dms\_SwiUimSelect ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_dms\_SwiUimSelect\_t \* *reqArg* )

Sets the DMS swi UIM Select parameters pack. A reset is required for any change in the UIM selection to take effect.

## 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

**9.5.2.62** `int pack_dms_UIMChangePIN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMChangePIN_t * pReq )`

To set UIM change PIN pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<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

**9.5.2.63** `int pack_dms_UIMGetControlKeyStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMGetControlKeyStatus_t * pReq )`

To set UIM Get Control Key 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
out	<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

**9.5.2.64** `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, 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

9.5.2.65 int pack\_dms\_UIMGetPINStatus ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

To set UIM Get PIN Status pack

## Parameters

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length

## Returns

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

## See Also

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

9.5.2.66 int pack\_dms\_UIMSetControlKeyProtection ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_dms\_UIMSetControlKeyProtection\_t \* pReq )

To set UIM Set Control Key Protection pack

## Parameters

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
out	pReq	request parameter

## Returns

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

## See Also

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

9.5.2.67 int pack\_dms\_UIMSetPINProtection ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_dms\_UIMSetPINProtection\_t \* pReq )

To set UIM PIN protection pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>pReq</i>	requeset parameter

**Returns**

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

**See Also**

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

**9.5.2.68** int pack\_dms\_UIMUnlockControlKey ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_UIMUnlockControlKey\_t \* *pReq* )

To set UIM Unblock Control Key pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>pReq</i>	requeset parameter

**Returns**

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

**See Also**

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

**9.5.2.69** int pack\_dms\_UIMUnlockPIN ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_dms\_UIMUnlockPIN\_t \* *pReq* )

To set UIM unblock PIN pack

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>pReq</i>	requeset parameter

**Returns**

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

**See Also**

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

9.5.2.70 `int pack_dms_UIMVerifyPIN ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_UIMVerifyPIN_t * pReq )`

To set UIM verify PIN pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<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

9.5.2.71 `int pack_dms_ValidateSPC ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_dms_ResetToFactoryDefaults_t * pReq )`

Requests the device to perform automatic service activation pack

#### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<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

9.5.2.72 `int unpack_dms_ActivateAutomatic ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

Requests the device to perform automatic service activation 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

9.5.2.73 `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

## Note

This is only supported for 3GPP2 devices.

9.5.2.74 `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

9.5.2.75 `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

**9.5.2.76** `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

**9.5.2.77** `int unpack_dms_GetCustFeaturesV2 ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetCustFeaturesV2_t * pOutput )`

Get Custom Feature 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

**9.5.2.78** `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

**9.5.2.79** 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

**9.5.2.80** int unpack\_dms\_GetDeviceCapabilitiesV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetDeviceCapabilitiesV2\_t \* *pOutput* )

Get device capability unpack v2

**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

**9.5.2.81** int unpack\_dms\_GetDeviceHardwareRev ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetDeviceHardwareRev\_t \* *pOutput* )

Get Hardware Revision of the device 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

9.5.2.82 `int unpack_dms_GetDeviceMfr ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetDeviceMfr_t * pOutput )`

Get Device Manufacture Name 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

9.5.2.83 `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

9.5.2.84 `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

**9.5.2.85** `int unpack_dms_GetFirmwareRevision ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetFirmwareRevision_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

**9.5.2.86** `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

**9.5.2.87** `int unpack_dms_GetFSN ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetFSN_t * pOutput )`

Get FSN unpack This API get the Factory Sequence Number of the device.

**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

#### 9.5.2.88 int unpack\_dms\_GetHardwareRevision ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetHardwareRevision\_t \* *pOutput* )

Get hardware revision of the device 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

#### 9.5.2.89 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

#### 9.5.2.90 int unpack\_dms\_GetManufacturer ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetManufacturer\_t \* *pOutput* )

To get device manufacturer information 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

9.5.2.91 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

9.5.2.92 int unpack\_dms\_GetNetworkTime ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetNetworkTime\_t \* *pOutput* )

Get Network Time unpack. Returns the current time of the device based on the value supported by the network.

## 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

9.5.2.93 int unpack\_dms\_GetNetworkTimeV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetNetworkTimeV2\_t \* *pOutput* )

Get Network Time unpack. Returns the current time of the device based on the value supported by the network.

## 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

9.5.2.94 int unpack\_dms\_GetOfflineReason ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetOfflineReason\_t \* *pOutput* )

To get operating mode offline reason 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

9.5.2.95 int unpack\_dms\_GetPower ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetPower\_t \* *pOutput* )

Get operating mode of the device 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

9.5.2.96 int unpack\_dms\_GetPRLVersion ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetPRLVersion\_t \* *pOutput* )

Get PRL Versions unpack Return version of the active Preferred Roaming List (PRL) in use by the device.

**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

**9.5.2.97** 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

**9.5.2.98** int unpack\_dms\_GetUSBComp ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_GetUSBComp\_t \* *pOutput* )

Get USB Comp unpack This API queries the modem's USB interface configuration and supported configuration parameters.

**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

9.5.2.99 `int unpack_dms_GetVoiceNumber ( uint8_t * pResp, uint16_t respLen, unpack_dms_GetVoiceNumber_t * pOutput )`

Get Voice Number unpack Returns the voice number in use by the device.

#### 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

9.5.2.100 `int unpack_dms_PSMCfgChange_ind ( uint8_t * pResp, uint16_t respLen, unpack_dms_PSMCfgChange_ind_t * pOutput )`

PSM Configuration 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

9.5.2.101 `int unpack_dms_ResetToFactoryDefaults ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

Resets to default factory settings of the device 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

9.5.2.102 `int unpack_dms_SetActivationStatusCallback ( uint8_t * pResp, uint16_t respLen,  
unpack_dms_SetActivationStatusCallback_t * pOutput )`

Set Activation 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

9.5.2.103 `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

9.5.2.104 `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

9.5.2.105 `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

9.5.2.106 `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

9.5.2.107 `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

**9.5.2.108** 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

**9.5.2.109** int unpack\_dms\_SetIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetIndication-  
Register\_t \* *pOutput* )

Set Indication 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

**9.5.2.110** int unpack\_dms\_SetPower ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SetPower\_t \* *pOutput* )

Set Power unpack. Sets the operating mode of the device.

**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

9.5.2.111 `int unpack_dms_SetUSBComp ( uint8_t * pResp, uint16_t respLen, unpack_dms_SetUSBComp_t * pOutput )`

Set USB Comp unpack 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

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

9.5.2.112 `int unpack_dms_SLQSDmsSwiGetPCInfo ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetPCInfo_t * pOutput )`

Get device power control status information 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

9.5.2.113 `int unpack_dms_SLQSDmsSwiGetResetInfo ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSDmsSwiGetResetInfo_t * pOutput )`

To get reset 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

9.5.2.114 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

9.5.2.115 int unpack\_dms\_SLQSDmsSwiGetUimSelection ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSDmsSwiGetUimSelection\_t \* *pOutput* )

Get device UIM selection information 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

9.5.2.116 `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

9.5.2.117 `int unpack_dms_SLQSGetBandCapability ( uint8_t * pResp, uint16_t respLen,  
unpack_dms_SLQSGetBandCapability_t * pOutput )`

Get band capability of the device 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

9.5.2.118 `int unpack_dms_SLQSGetBandCapabilityExt ( uint8_t * pResp, uint16_t respLen,  
unpack_dms_SLQSGetBandCapabilityExt_t * pOutput )`

Get band capability of the device 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

9.5.2.119 `int unpack_dms_SLQSGetERIFile ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSGetERIFile_t * pOutput )`

To get UIM Get ERI file 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

9.5.2.120 `int unpack_dms_SLQSGetPowerSaveModeConfig ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSGetPowerSaveModeConfig_t * pOutput )`

Get Power Save Mode (PSM) configuration parameter 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

9.5.2.121 `int unpack_dms_SLQSSetPowerSaveModeConfig ( uint8_t * pResp, uint16_t respLen, unpack_dms_SLQSSetPowerSaveModeConfig_t * pOutput )`

Sets the Power Save Mode (PSM) configuration parameters 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

**9.5.2.122** 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

**9.5.2.123** int unpack\_dms\_SLQSSwiGetCrashInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSSwiGet-  
CrashInfo\_t \* *pOutput* )

To Get the Crash Information from the device 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

**9.5.2.124** int unpack\_dms\_SLQSSwiGetDyingGaspCfg ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t \* *pOutput* )

Get 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

9.5.2.125 int unpack\_dms\_SLQSSwiGetDyingGaspStatistics ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t \* *pOutput* )

Get 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

9.5.2.126 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

9.5.2.127 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

9.5.2.128 int unpack\_dms\_SLQSSwiGetHostDevInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetHostDevInfo\_t \* *pOutput* )

To get host dev information 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

9.5.2.129 int unpack\_dms\_SLQSSwiGetOSInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSSwiGetOSInfo\_t  
\* *pOutput* )

To get Host OS 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

9.5.2.130 int unpack\_dms\_SLQSSwiGetSerialNoExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiGetSerialNoExt\_t \* *pOutput* )

To get serial number extension 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

9.5.2.131 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

9.5.2.132 int unpack\_dms\_SLQSSwiSetHostDevInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_dms\_SLQSSwiSetHostDevInfo\_t \* *pOutput* )

To set Host Dev 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

9.5.2.133 int unpack\_dms\_SLQSSwiSetOSInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSSwiSetOSInfo\_t  
\* *pOutput* )

To set Host OS 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

**9.5.2.134** int unpack\_dms\_SLQSUIGetState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SLQSUIGetState\_t \* *pOutput* )

Returns the UIM state unpack. This API is deprecated on MC73xx/EM73xx modules since firmware version SW-I9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use API [unpack\\_uim\\_GetCardStatus](#) / [unpack\\_uim\\_GetCardStatus](#) 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

**9.5.2.135** int unpack\_dms\_SwiEventReportCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SwiEventReportCallBack\_ind\_t \* *pOutput* )

unpack for DMS swi event report indication

## 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

**9.5.2.136** int unpack\_dms\_SwiSetEventReport ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SwiSetEventReport\_t \* *pOutput* )

Sets the swi event report parameters unpack

## 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

9.5.2.137 int unpack\_dms\_SwiUimSelect ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_SwiUimSelect\_t \* *pOutput* )

Sets the DMS swi UIM Select parameters unpack

## 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

9.5.2.138 int unpack\_dms\_UIMChangePIN ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMSetPINProtection\_t \* *pOutput* )

To get UIM 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

9.5.2.139 int unpack\_dms\_UIMGetControlKeyStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMGetControlKeyStatus\_t \* *pOutput* )

To get UIM Get ControlKeyStatus 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

9.5.2.140 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

9.5.2.141 int unpack\_dms\_UIMGetPINStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMGetPINStatus\_t \* *pOutput* )

To get UIM Get PIN 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

9.5.2.142 int unpack\_dms\_UIMSetControlKeyProtection ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMSetControlKeyProtection\_t \* *pOutput* )

To get UIM Set Control Key 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

**9.5.2.143** int unpack\_dms\_UIMSetPINProtection ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMSetPIN-Protection\_t \* *pOutput* )

To get UIM 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

**9.5.2.144** int unpack\_dms\_UIMUnblockControlKey ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_dms\_UIMUnblock-ControlKey\_t \* *pOutput* )

To get UIM Unblock Control Key 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

9.5.2.145 `int unpack_dms_UIMUnblockPIN ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t * pOutput )`

To get UIM 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

9.5.2.146 `int unpack_dms_UIMVerifyPIN ( uint8_t * pResp, uint16_t respLen, unpack_dms_UIMSetPINProtection_t * pOutput )`

To get UIM 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

9.5.2.147 `int unpack_dms_ValidateSPC ( uint8_t * pResp, uint16_t respLen, unpack_dms_ResetToFactoryDefaults_t * pOutput )`

Requests the device to perform automatic service activation 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 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\\_SIZE 16](#)
- [#define LITE\\_TOTAL\\_IMAGE\\_ID\\_STRING\\_SIZE 100](#)
- [#define FMS\\_GOBI\\_MBN\\_IMG\\_ID\\_STR\\_LEN LITE\\_TOTAL\\_IMAGE\\_ID\\_STRING\\_SIZE](#)
- [#define FMS\\_GOBI\\_MBN\\_BUILD\\_ID\\_STR\\_LEN LITE\\_TOTAL\\_IMAGE\\_ID\\_STRING\\_SIZE](#)
- [#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)



### 9.6.1 Macro Definition Documentation

9.6.1.1 `#define FMS_FW_PRI_BUILD_MATCH_LEN 11`

9.6.1.2 `#define FMS_GOBI_LISTENTRIES_MAX 2`

9.6.1.3 `#define FMS_GOBI_MBN_BUILD_ID_STR_LEN LITE_TOTAL_IMAGE_ID_STRING_SIZE`

9.6.1.4 `#define FMS_GOBI_MBN_IMG_ID_SIZE 16`

9.6.1.5 `#define FMS_GOBI_MBN_IMG_ID_STR_LEN LITE_TOTAL_IMAGE_ID_STRING_SIZE`

9.6.1.6 `#define FMS_IMAGE_ID_BUILD_ID_LEN 32`

9.6.1.7 `#define FMS_IMAGE_ID_IMG_ID_LEN 16`

9.6.1.8 `#define FMS_IMAGE_ID_MAX_ENTRIES 2`

9.6.1.9 `#define FMS_IMAGE_ID_PRI_IMGTYPE 0x01`

9.6.1.10 `#define FMS_MAX_IMAGE_ID_ELEMENT 50`

9.6.1.11 `#define FMS_MAX_IMAGE_PREFERENCE_IMAGE_SIZE 255`

9.6.1.12 `#define LITE_TOTAL_IMAGE_ID_STRING_SIZE 100`

### 9.6.2 Function Documentation

9.6.2.1 `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 GetStoredImages</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>pValidCombinations</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

```
9.6.2.2 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, 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

```
9.6.2.3 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, 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

```
9.6.2.4 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, 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

```
9.6.2.5 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

```
9.6.2.6 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

9.6.2.7 `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

## 9.7 ims.h File Reference

### Data Structures

- struct [pack\\_ims\\_SLQSSetSIPConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSSetSIPConfig\\_t](#)
- struct [pack\\_ims\\_SLQSSetRegMgrConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSSetRegMgrConfig\\_t](#)
- struct [pack\\_ims\\_SLQSSetIMSSMSCConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSSetIMSSMSCConfig\\_t](#)
- struct [pack\\_ims\\_SLQSSetIMSUserConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSSetIMSUserConfig\\_t](#)
- struct [pack\\_ims\\_SLQSSetIMSVoIPConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSSetIMSVoIPConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSGetSIPConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSGetRegMgrConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSGetIMSSMSCConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSGetIMSUserConfig\\_t](#)
- struct [unpack\\_ims\\_SLQSGetIMSVoIPConfig\\_t](#)
- struct [pack\\_ims\\_SLQSImsConfigIndicationRegister\\_t](#)
- struct [ims\\_PCSCFPortInfo](#)
- struct [ims\\_CSCFPortNameInfo](#)
- struct [ims\\_IMSTestModelInfo](#)
- struct [unpack\\_ims\\_SLQSRegMgrCfgCallBack\\_ind\\_t](#)
- struct [ims\\_SIPPortInfo](#)
- struct [ims\\_SIPRegnTmrInfo](#)
- struct [ims\\_SubscrTmrInfo](#)
- struct [ims\\_TmrT1Info](#)
- struct [ims\\_TmrT2Info](#)
- struct [ims\\_TmrTflInfo](#)
- struct [ims\\_SigCompEnInfo](#)
- struct [unpack\\_ims\\_SLQSSIPCfgCallBack\\_ind\\_t](#)
- struct [ims\\_SMSFmtInfo](#)
- struct [ims\\_SMSoIPNwInfo](#)
- struct [ims\\_PhCtxtURIInfo](#)
- struct [unpack\\_ims\\_SLQSSMSCfgCallBack\\_ind\\_t](#)
- struct [ims\\_IMSDomainInfo](#)

- struct [unpack\\_ims\\_SLQSSetSIPConfig\\_ind\\_t](#)
- struct [ims\\_SessDurInfo](#)
- struct [ims\\_MinSessExpInfo](#)
- struct [ims\\_EnabAMRWBInfo](#)
- struct [ims\\_EnabSCRAMRInfo](#)
- struct [ims\\_EnabSCRAMRWBInfo](#)
- struct [ims\\_AMRModelInfo](#)
- struct [ims\\_AMRWBModelInfo](#)
- struct [ims\\_AMROctAlgnInfo](#)
- struct [ims\\_AMRWBOctAlgnInfo](#)
- struct [ims\\_RngTmrInfo](#)
- struct [ims\\_RngBkTmrInfo](#)
- struct [ims\\_RTPRTCPInactTmrDurInfo](#)
- struct [unpack\\_ims\\_SLQSVolIPCfgCallback\\_ind\\_t](#)

## Macros

- `#define MAX_NAME_LEN 255`

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_ims\\_SLQSImsConfigIndicationRegister\\_t](#)

## Functions

- int [pack\\_ims\\_SLQSSetSIPConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSSetSIPConfig\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSSetSIPConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSetSIPConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSSetRegMgrConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSSetRegMgrConfig\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSSetRegMgrConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSetRegMgrConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSSetIMSSMSConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSSetIMSSMSConfig\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSSetIMSSMSConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSetIMSSMSConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSSetIMSUserConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSSetIMSUserConfig\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSSetIMSUserConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSetIMSUserConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSSetIMSVoIPConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSSetIMSVoIPConfig\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSSetIMSVoIPConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSetIMSVoIPConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSGetSIPConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_ims\\_SLQSGetSIPConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSGetSIPConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSGetRegMgrConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_ims\\_SLQSGetRegMgrConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSGetRegMgrConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSGetIMSSMSConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_ims\\_SLQSGetIMSSMSConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSGetIMSSMSConfig\\_t](#) \*pOutput)

- int [pack\\_ims\\_SLQSGetIMSUserConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_ims\\_SLQSGetIMSUserConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSGetIMSUserConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSGetIMSVoIPConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_ims\\_SLQSGetIMSVoIPConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSGetIMSVoIPConfig\\_t](#) \*pOutput)
- int [pack\\_ims\\_SLQSImsConfigIndicationRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_ims\\_SLQSImsConfigIndicationRegister\\_t](#) \*pReqParam)
- int [unpack\\_ims\\_SLQSImsConfigIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSImsConfigIndicationRegister\\_t](#) \*pOutput)
- int [unpack\\_ims\\_SLQSRegMgrCfgCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSRegMgrCfgCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_ims\\_SLQSSIPCfgCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSIPCfgCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_ims\\_SLQSSMSCfgCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSSMSCfgCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_ims\\_SLQSUserCfgCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSUserCfgCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_ims\\_SLQSVoIPCfgCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_ims\\_SLQSVoIPCfgCallBack\\_ind\\_t](#) \*pOutput)

### 9.7.1 Macro Definition Documentation

9.7.1.1 `#define MAX_NAME_LEN 255`

### 9.7.2 Typedef Documentation

9.7.2.1 `typedef unpack_result_t unpack_ims_SLQSImsConfigIndicationRegister_t`

### 9.7.3 Function Documentation

9.7.3.1 `int pack_ims_SLQSGetIMSSMSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Retrieves the SMS configuration parameters for the requesting control point 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

Device Supported: MC73xx, MC74xx and EM74xx

9.7.3.2 `int pack_ims_SLQSGetIMSUserConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Retrieves the IMS User configuration parameters for the requesting control point 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**

Device Supported: MC73xx, MC74xx and EM74xx

### 9.7.3.3 int pack\_ims\_SLQSGetIMSVoIPConfig ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Retrieves the IMS VoIP configuration parameters for the requesting control point 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**

Device Supported: MC73xx, MC74xx and EM74xx

### 9.7.3.4 int pack\_ims\_SLQSGetRegMgrConfig ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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

Device Supported: MC73xx, MC74xx and EM74xx

### 9.7.3.5 int pack\_ims\_SLQSGetSIPConfig ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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

Device Supported: MC73xx, MC74xx and EM74xx

### 9.7.3.6 int pack\_ims\_SLQSImsConfigIndicationRegister ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_ims\_SLQSImsConfigIndicationRegister\_t \* pReqParam )

Sets the registration state for different QMI\_IMS indications for the requesting control point 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>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

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

- This function 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.

**9.7.3.7** `int pack_ims_SLQSSetIMSSMSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetIMSSMSConfig_t * pReqParam )`

Sets the IMS SMS configuration parameters for the requesting control point 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>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

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

**9.7.3.8** `int pack_ims_SLQSSetIMSUserConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetIMSUserConfig_t * pReqParam )`

Sets the IMS user configuration parameters for the requesting control point 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>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



**Note**

Device Supported: MC73xx, MC74xx and EM74xx

**9.7.3.9** `int pack_ims_SLQSSetIMSVoIPConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetIMSVoIPConfig_t * pReqParam )`

Sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point 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>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

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

**9.7.3.10** `int pack_ims_SLQSSetRegMgrConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetRegMgrConfig_t * pReqParam )`

Sets the IMS registration manager configuration parameters for the requesting control point 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>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

**Note**

Device Supported: MC73xx, MC74xx and EM74xx

**9.7.3.11** `int pack_ims_SLQSSetSIPConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_ims_SLQSSetSIPConfig_t * pReqParam )`

Sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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>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

#### Note

Device Supported: MC73xx, MC74xx and EM74xx

**9.7.3.12** `int unpack_ims_SLQSGetIMSSMSConfig ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSGetIMSSMSConfig_t * pOutput )`

Retrieves the SMS configuration parameters for the requesting control point 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

**9.7.3.13** `int unpack_ims_SLQSGetIMSUserConfig ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSGetIMSUserConfig_t * pOutput )`

Retrieves the IMS User configuration parameters for the requesting control point 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

#### 9.7.3.14 int unpack\_ims\_SLQSGetIMSVolPConfig ( uint8\_t \* pResp, uint16\_t respLen, unpack\_ims\_SLQSGetIMSVolPConfig\_t \* pOutput )

Retrieves the IMS VoIP configuration parameters for the requesting control point 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

#### 9.7.3.15 int unpack\_ims\_SLQSGetRegMgrConfig ( uint8\_t \* pResp, uint16\_t respLen, unpack\_ims\_SLQSGetRegMgrConfig\_t \* pOutput )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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

#### 9.7.3.16 int unpack\_ims\_SLQSGetSIPConfig ( uint8\_t \* pResp, uint16\_t respLen, unpack\_ims\_SLQSGetSIPConfig\_t \* pOutput )

Retrieves the Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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

**9.7.3.17** int unpack\_ims\_SLQSImsConfigIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_ims\_SLQSImsConfigIndicationRegister\_t \* *pOutput* )

Sets the registration state for different QMI\_IMS indications for the requesting control point 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

**9.7.3.18** int unpack\_ims\_SLQSRegMgrCfgCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t \* *pOutput* )

Unpack indication about IMS registration manager configuration

**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

**9.7.3.19** int unpack\_ims\_SLQSSetIMSSMSConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_ims\_SLQSSetIMSSMS-  
Config\_t \* *pOutput* )

Sets the IMS SMS configuration parameters for the requesting control point 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

### 9.7.3.20 int unpack\_ims\_SLQSSetIMSUserConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_ims\_SLQSSetIMSUserConfig\_t \* *pOutput* )

Sets the IMS user configuration parameters for the requesting control point 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

### 9.7.3.21 int unpack\_ims\_SLQSSetIMSVoIPConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_ims\_SLQSSetIMSVoIPConfig\_t \* *pOutput* )

Sets the IMS Voice over Internet Protocol (VoIP) configuration parameters for the requesting control point 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

### 9.7.3.22 int unpack\_ims\_SLQSSetRegMgrConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_ims\_SLQSSetRegMgrConfig\_t \* *pOutput* )

Sets the IMS registration manager configuration parameters for the requesting control point 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

**9.7.3.23** `int unpack_ims_SLQSSetSIPConfig ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSSetSIPConfig_t * pOutput )`

Sets the IMS Session Initiation Protocol(SIP) configuration parameters for the requesting control point 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

**9.7.3.24** `int unpack_ims_SLQSSIPCfgCallBack_ind ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSSIPCfgCallBack_ind_t * pOutput )`

Unpack indication about SIP configuration info

**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

**9.7.3.25** `int unpack_ims_SLQSSMSCfgCallBack_ind ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSSMSCfgCallBack_ind_t * pOutput )`

Unpack indication about SMS configuration info for IMS

**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

**9.7.3.26** `int unpack_ims_SLQSTUserCfgCallBack_ind ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSTUserCfgCallBack_ind_t * pOutput )`

Unpack indication about User configuration info for IMS

## 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

**9.7.3.27** `int unpack_ims_SLQSVolPCfgCallBack_ind ( uint8_t * pResp, uint16_t respLen, unpack_ims_SLQSVolPCfgCallBack_ind_t * pOutput )`

Unpack indication about VOIP configuration info

## 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

## 9.8 imsa.h File Reference

### Data Structures

- struct [pack\\_imsa\\_SLQSTRegisterIMSAIndication\\_t](#)
- struct [unpack\\_imsa\\_SLQSTGetIMSARegStatus\\_t](#)
- struct [unpack\\_imsa\\_SLQSTGetIMSAServiceStatus\\_t](#)

- struct [imsa\\_SmsSvcStatusInfo](#)
- struct [imsa\\_SmsRatInfo](#)
- struct [imsa\\_VoipSvcStatusInfo](#)
- struct [imsa\\_VoipRatInfo](#)
- struct [imsa\\_VtSvcStatusInfo](#)
- struct [imsa\\_VtRatInfo](#)
- struct [imsa\\_UtSvcStatusInfo](#)
- struct [imsa\\_UtRatInfo](#)
- struct [unpack\\_imsa\\_SLQSImsaSvcStatusCallBack\\_ind\\_t](#)
- struct [imsa\\_IMSRegStatusInfo](#)
- struct [imsa\\_IMSRegStatusErrorCodeInfo](#)
- struct [imsa\\_NewIMSRegStatusInfo](#)
- struct [unpack\\_imsa\\_SLQSImsaRegStatusCallBack\\_ind\\_t](#)
- struct [imsa\\_RatHandoverStatusInfo](#)
- struct [unpack\\_imsa\\_SLQSImsaRatStatusCallBack\\_ind\\_t](#)
- struct [imsa\\_IMSFailErrCodeTlv](#)
- struct [unpack\\_imsa\\_SLQSImsaPdpStatusCallBack\\_ind\\_t](#)

## Macros

- `#define MAX_ERROR_CODE_LEN 256`

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_imsa\\_SLQSRegisterIMSAIndication\\_t](#)

## Functions

- int [pack\\_imsa\\_SLQSRegisterIMSAIndication](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_imsa\\_SLQSRegisterIMSAIndication\\_t](#) \*reqParam)
- int [unpack\\_imsa\\_SLQSRegisterIMSAIndication](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSRegisterIMSAIndication\\_t](#) \*pOutput)
- int [pack\\_imsa\\_SLQSGetIMSARegStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- int [unpack\\_imsa\\_SLQSGetIMSARegStatus](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSGetIMSARegStatus\\_t](#) \*pOutput)
- int [pack\\_imsa\\_SLQSGetIMSAServiceStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- int [unpack\\_imsa\\_SLQSGetIMSAServiceStatus](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSGetIMSAServiceStatus\\_t](#) \*pOutput)
- int [unpack\\_imsa\\_SLQSImsaSvcStatusCallBack\\_ind](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSImsaSvcStatusCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_imsa\\_SLQSImsaRegStatusCallBack\\_ind](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSImsaRegStatusCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_imsa\\_SLQSImsaRatStatusCallBack\\_ind](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSImsaRatStatusCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_imsa\\_SLQSImsaPdpStatusCallBack\\_ind](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_imsa\\_SLQSImsaPdpStatusCallBack\\_ind\\_t](#) \*pOutput)



### 9.8.1 Macro Definition Documentation

9.8.1.1 `#define MAX_ERROR_CODE_LEN 256`

### 9.8.2 Typedef Documentation

9.8.2.1 `typedef unpack_result_t unpack_imsa_SLQSRegisterIMSAIndication_t`

### 9.8.3 Function Documentation

9.8.3.1 `int pack_imsa_SLQSGetIMSARegStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack Get IMSA Registration Status command.

#### 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

9.8.3.2 `int pack_imsa_SLQSGetIMSAServiceStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack Get IMSA Service Status command

#### 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

**9.8.3.3** `int pack_imsa_SLQSRegisterIMSAIndication ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_imsa_SLQSRegisterIMSAIndication_t * reqParam )`

Function to pack Register IMSA Indication command

## 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_imsa_SLQSRegisterIMSAIndication_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

**9.8.3.4** `int unpack_imsa_SLQSGetIMSARegStatus ( uint8_t * pResp, uint16_t respLen, unpack_imsa_SLQSGetIMSARegStatus_t * pOutput )`

Function to unpack Get IMSA Registration Status response from modem. \*

## 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_imsa_SLQSGetIMSARegStatus_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

**9.8.3.5** `int unpack_imsa_SLQSGetIMSAServiceStatus ( uint8_t * pResp, uint16_t respLen, unpack_imsa_SLQSGetIMSAServiceStatus_t * pOutput )`

Function to unpack Get IMSA Service Status response from modem

## Parameters

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • See <a href="#">unpack_imsa_SLQSGetIMSAServiceStatus_t</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

**9.8.3.6** `int unpack_imsa_SLQSImsaPdpStatusCallBack_ind ( uint8_t * pResp, uint16_t respLen, unpack_imsa_SLQSImsaPdpStatusCallBack_ind_t * pOutput )`

Unpack indication about IMSA PDP status info

## 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

```
9.8.3.7 int unpack_imsa_SLQSImsaRatStatusCallBack_ind ( uint8_t * pResp, uint16_t respLen,
unpack_imsa_SLQSImsaRatStatusCallBack_ind_t * pOutput )
```

Unpack indication about IMSA RAT status info

#### 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

```
9.8.3.8 int unpack_imsa_SLQSImsaRegStatusCallBack_ind ( uint8_t * pResp, uint16_t respLen,
unpack_imsa_SLQSImsaRegStatusCallBack_ind_t * pOutput )
```

Unpack indication about IMSA registration status info

#### 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

```
9.8.3.9 int unpack_imsa_SLQSImsaSvcStatusCallBack_ind ( uint8_t * pResp, uint16_t respLen,
unpack_imsa_SLQSImsaSvcStatusCallBack_ind_t * pOutput )
```

Unpack indication about IMSA service status info

#### 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

9.8.3.10 `int unpack_imsa_SLQSRegisterIMSAIndication ( uint8_t * pResp, uint16_t respLen, unpack_imsa_SLQSRegisterIMSAIndication_t * pOutput )`

Function to unpack Register IMSA Indication response from modem

## Parameters

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • See <a href="#">unpack_imsa_SLQSRegisterIMSAIndication_t</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

## 9.9 lite-fw.h File Reference

### Data Structures

- [struct \\_litefw\\_FirmwareInfo\\_](#)
- [struct \\_litefw\\_FirmwarePartNo\\_](#)
- [struct \\_litefw\\_FirmwareFileInfo](#)

### Macros

- `#define IMG_MASK_MDM (1<<0)`
- `#define IMG_MASK_PRI (1<<1)`
- `#define IMG_MASK_CLEAR (0x000)`
- `#define IMG_MASK_GENERIC (0xFFFF)`
- `#define LITEFW_CARRIER_PACKAGE_SKU "9999999"`
- `#define LITEFW_SKU_STRING_LENGTH 7`
- `#define FIRMWARE_INFO_STRING_SIZE 20`
- `#define NULL_TERMINATOR_CHAR_SIZE 1`
- `#define FIRMWARE_IMAGE_SIZE_MAX 512`
- `#define FIRMWARE_BCHVERSTRSIZE 84`
- `#define FIRMWARE_BCHDATESIZE 8`
- `#define MAX_IMAGE_PRODUCT_LENGTH 10`
- `#define libSDP_GetVersion litefw_GetVersion`
- `#define libSDP_CalculateImageMask litefw_CalculateImageMask`

- `#define libSDP_getFileType litefw_getFileType`
- `#define libSDP_ExtractFirmwareParametersByPath litefw_ExtractFirmwareParametersByPath`
- `#define libSDP_GetModelFamily litefw_GetModelFamily`
- `#define libSDP_CheckValidFirmwareInfo litefw_CheckValidFirmwareInfo`
- `#define libSDP_BuildImagesPreferenceRequest litefw_BuildImagesPreferenceRequest`
- `#define libSDP_DownloadFW litefw_DownloadFW`
- `#define libsdp_set_log_func litefw_set_log_func`
- `#define libsdp_SetReadBlockSize litefw_SetReadBlockSize`
- `#define LIBSDP_CARRIER_PACKAGE_SKU LITEFW_CARRIER_PACKAGE_SKU`
- `#define LIBSDP_SKU_STRING_LENGTH LITEFW_SKU_STRING_LENGTH`

## Typedefs

- typedef struct  
    [\\_litefw\\_FirmwareInfo](#) [litefw\\_FirmwareInfo](#)
- typedef struct  
    [\\_litefw\\_FirmwarePartNo](#) [litefw\\_FirmwarePartNo](#)
- typedef struct  
    [\\_litefw\\_FirmwareFileInfo](#) [litefw\\_FirmwareFileInfo](#)
- typedef void(\* [litefwlogger](#) )(uint8\_t lvl, const char \*buff)
- typedef [litefw\\_FirmwareInfo](#) [libSDP\\_FirmwareInfo](#)
- typedef [litefwlogger](#) [libsdplogger](#)

## Enumerations

- enum [litefw\\_fwdwl\\_error\\_codes](#) {  
    [eSDP\\_FWDWL\\_SUCCESS](#) = 0,  
    [eSDP\\_FWDWL\\_ERR\\_GENERAL](#) = 100,  
    [eSDP\\_FWDWL\\_ERR\\_SDK](#),  
    [eSDP\\_FWDWL\\_ERR\\_SET\\_CBK](#),  
    [eSDP\\_FWDWL\\_ERR\\_PATH\\_TOO\\_LONG](#),  
    [eSDP\\_FWDWL\\_ERR\\_PATH\\_NOT\\_SPECIFIED](#),  
    [eSDP\\_FWDWL\\_ERR\\_FW\\_UPGRADE](#),  
    [eSDP\\_FWDWL\\_ERR\\_INVALID\\_DEV](#),  
    [eSDP\\_FWDWL\\_ERR\\_INVALID\\_PATH](#),  
    [eSDP\\_FWDWL\\_ERR\\_TIMEOUT](#),  
    [eSDP\\_FWDWL\\_ERR\\_FAIL](#),  
    [eSDP\\_FWDWL\\_ERR\\_PRI\\_FAIL](#),  
    [eSDP\\_FWDWL\\_ERR\\_FW\\_VERSION\\_FAIL](#),  
    [eSDP\\_FWDWL\\_ERR\\_SDP\\_TIMEOUT](#),  
    [eFIREHOSE\\_ERR\\_SECBOOT\\_INVALID\\_CERT\\_CHAIN](#),  
    [eSDP\\_FWDWL\\_ERR\\_END](#) }
- enum [litefw\\_Fw\\_Type](#) {  
    [eFW\\_TYPE\\_MBN\\_GOBI](#),  
    [eFW\\_TYPE\\_MBN](#),  
    [eFW\\_TYPE\\_CWE](#),  
    [eFW\\_TYPE\\_NVU](#),  
    [eFW\\_TYPE\\_SPK](#),  
    [eFW\\_TYPE\\_INVALID](#),  
    [eFW\\_TYPE\\_CWE\\_NVU](#) }
- enum [litefw\\_fileimgtype](#) {  
    [eFILE\\_TYPE\\_NONE](#) = 0,  
    [eFILE\\_TYPE\\_CAR\\_PRI](#),  
    [eFILE\\_TYPE\\_OEM\\_PRI](#),  
    [eFILE\\_TYPE\\_COMPO\\_PRI](#) }

- enum `litefw_imagetype` {  
`eIMAGE_TYPE_MIN` = 0,  
`eIMAGE_TYPE_BOOT` = `eIMAGE_TYPE_MIN`,  
`eIMAGE_TYPE_APPL`,  
`eIMAGE_TYPE_APPS`,  
`eIMAGE_TYPE_FILE`,  
`eIMAGE_TYPE_SPKG`,  
`eIMAGE_TYPE_MODM`,  
`eIMAGE_TYPE_USER`,  
`eIMAGE_TYPE_KEYS`,  
`eIMAGE_TYPE_MAX`,  
`eIMAGE_TYPE_INVALID` = 0xFF,  
`eIMAGE_TYPE_ANY` = `eIMAGE_TYPE_INVALID` }
- enum `litefw_Models` {  
`eModel_Unknown` = -1,  
`eModel_9X15` = 0,  
`eModel_WP9X15` = 1,  
`eModel_9X30` = 2,  
`eModel_9x07` = 3,  
`eModel_9x06` = 4 }
- enum `litefw_QDL_MODEs` {  
`eQDL_MODE_Unknown` = -1,  
`eQDL_MODE_INIT` = 0,  
`eQDL_MODE_TTYUSB` = 1,  
`eQDL_MODE_UART0` = 2 }
- enum `litefw_QDL_FLOW_CONTROLS` {  
`eQDL_HW_FLOW_Unknown` = -1,  
`eQDL_HW_FLOW_INIT` = 0,  
`eQDL_HW_FLOW_ENABLE` = 1,  
`eQDL_HW_FLOW_DISABLE` = 2 }

## Functions

- void `litefw_logsenable` (int log\_en)
- int `litefw_ExtractFirmwareParametersByPath` (char \*pImagePath, `litefw_FirmwareInfo` \*info)
- int `litefw_ExtractFirmwarePartNoByPath` (char \*pImagePath, `litefw_FirmwarePartNo` \*partno)
- int `litefw_BuildImagesPreferenceRequest` (`litefw_FirmwareInfo` info, `pack_fms_SetImagesPreference_t` \*pack)
- int `litefw_CalculateImageMask` (`unpack_fms_SetImagesPreference_t` SetPrefRespFromModem)
- int `litefw_getFileType` (char \*szPath)
- unsigned int `litefw_DownloadFW` (char \*pImagePath, char \*szTTYPath, int iFWImageType, int image\_mask, int iModelFamily)
- int `litefw_GetModelFamily` (char \*pModelString)
- int `litefw_CheckValidFirmwareInfo` (`litefw_FirmwareInfo` info)
- char \* `litefw_GetVersion` ()
- int `litefw_set_log_func` (`litefwlogger` \*func)
- void `litefw_SetReadBlockSize` (unsigned long IBlockSize)
- int `litefw_switch_9x07_to_downloadmode` (char \*szTTYPath)
- int `litefw_switch_to_BootHoldMode` (char \*szTTYPath)
- int `litefw_SetQTLDownloadMode` (int iMode)
- int `litefw_GetQTLDownloadMode` ()
- int `litefw_SetQTLHWFlowControl` (int iMode)
- int `litefw_GetQTLHWFlowControl` ()
- enum `eQCWWANError` `litefw_SLQSGetFirmwareFileInfo` (char \*fullFilePath, `litefw_FirmwareFileInfo` \*info, uint8\_t \*pNumOfItems)

### 9.9.1 Detailed Description

Filename: [lite-fw.h](#)

Purpose: Global definitions used inside the SDK

Copyright: © 2016 Sierra Wireless Inc., all rights reserved

### 9.9.2 Macro Definition Documentation

9.9.2.1 `#define FIRMWARE_BCHDATESIZE 8`

9.9.2.2 `#define FIRMWARE_BCHVERSTRSIZE 84`

9.9.2.3 `#define FIRMWARE_IMAGE_SIZE_MAX 512`

9.9.2.4 `#define FIRMWARE_INFO_STRING_SIZE 20`

9.9.2.5 `#define IMG_MASK_CLEAR (0x000)`

9.9.2.6 `#define IMG_MASK_GENERIC (0xFFFF)`

9.9.2.7 `#define IMG_MASK_MDM (1<<0)`

9.9.2.8 `#define IMG_MASK_PRI (1<<1)`

9.9.2.9 `#define libSDP_BuildImagesPreferenceRequest litefw_BuildImagesPreferenceRequest`

9.9.2.10 `#define libSDP_CalculateImageMask litefw_CalculateImageMask`

9.9.2.11 `#define LIBSDP_CARRIER_PACKAGE_SKU LITEFW_CARRIER_PACKAGE_SKU`

9.9.2.12 `#define libSDP_CheckValidFirmwareInfo litefw_CheckValidFirmwareInfo`

9.9.2.13 `#define libSDP_DownloadFW litefw_DownloadFW`

9.9.2.14 `#define libSDP_ExtractFirmwareParametersByPath litefw_ExtractFirmwareParametersByPath`

9.9.2.15 `#define libSDP_getFileType litefw_getFileType`

9.9.2.16 `#define libSDP_GetModelFamily litefw_GetModelFamily`

9.9.2.17 `#define libSDP_GetVersion litefw_GetVersion`

9.9.2.18 `#define libsdp_set_log_func litefw_set_log_func`

9.9.2.19 `#define libsdp_SetReadBlockSize litefw_SetReadBlockSize`

9.9.2.20 `#define LIBSDP_SKU_STRING_LENGTH LITEFW_SKU_STRING_LENGTH`

9.9.2.21 `#define LITEFW_CARRIER_PACKAGE_SKU "9999999"`

9.9.2.22 `#define LITEFW_SKU_STRING_LENGTH 7`

9.9.2.23 `#define MAX_IMAGE_PRODUCT_LENGTH 10`



9.9.2.24 `#define NULL_TERMINATOR_CHAR_SIZE 1`

### 9.9.3 Typedef Documentation

9.9.3.1 `typedef litefw_FirmwareInfo libSDP_FirmwareInfo`

9.9.3.2 `typedef litefwlogger libsdplogger`

9.9.3.3 `typedef struct _litefw_FirmwareFileInfo litefw_FirmwareFileInfo`

This structure provided more detailed information of the provided firmware file

#### Parameters

<i>fullPath</i>	<ul style="list-style-type: none"> <li>• full path of the file</li> </ul>
<i>imagemask</i>	<ul style="list-style-type: none"> <li>• Bitmask provides type of file <ul style="list-style-type: none"> <li>– bit0 - cwe</li> <li>– bit1 - nvu</li> <li>– bit2 - spk</li> </ul> </li> </ul>
<i>headerType</i>	<ul style="list-style-type: none"> <li>• see <a href="#">litefw_imagetype</a></li> </ul>
<i>imageType</i>	<ul style="list-style-type: none"> <li>• see <a href="#">litefw_fileimgtype</a></li> </ul>
<i>modelIdStr</i>	<ul style="list-style-type: none"> <li>• device model id</li> </ul>
<i>partNoStr</i>	<ul style="list-style-type: none"> <li>• part number id</li> </ul>
<i>skuStr</i>	<ul style="list-style-type: none"> <li>• sku id</li> </ul>
<i>packageIdStr</i>	<ul style="list-style-type: none"> <li>• cwe sierra package id</li> </ul>
<i>carrierStr</i>	<ul style="list-style-type: none"> <li>• carrier id</li> </ul>
<i>priVersionStr</i>	<ul style="list-style-type: none"> <li>• pri version</li> </ul>
<i>versionStr</i>	<ul style="list-style-type: none"> <li>• firmware version</li> </ul>
<i>releaseDate</i>	<ul style="list-style-type: none"> <li>• release date of the file</li> </ul>

#### 9.9.3.4 typedef struct \_litedfw\_FirmwareInfo\_litedfw\_FirmwareInfo

This structure contains information of the provided firmware file

##### Parameters

<i>szModelid_str</i>	<ul style="list-style-type: none"> <li>Model Name String</li> </ul>
<i>szFwversion_str</i>	<ul style="list-style-type: none"> <li>Firmware Version String.</li> </ul>
<i>szSku_str</i>	<ul style="list-style-type: none"> <li>SKU String.</li> </ul>
<i>szPackageid_str</i>	<ul style="list-style-type: none"> <li>Package ID String.</li> </ul>
<i>szCarrier_str</i>	<ul style="list-style-type: none"> <li>Carrier String.</li> </ul>
<i>szCarrier-Priversion_str</i>	<ul style="list-style-type: none"> <li>Carrier PRI Version String.</li> </ul>

#### 9.9.3.5 typedef struct \_litedfw\_FirmwarePartNo\_litedfw\_FirmwarePartNo

This structure contains information of the response parameters associated with a Read Transparent API.

##### Parameters

<i>szPartno_str</i>	<ul style="list-style-type: none"> <li>Part Number of the provided firmware image</li> </ul>
---------------------	--

#### 9.9.3.6 typedef void(\* litedfwlogger)(uint8\_t lvl, const char \*buff)

This Custom Log prototype.

##### Parameters

in	<i>lvl</i>	<ul style="list-style-type: none"> <li>Log level.</li> </ul>
in	<i>buff</i>	<ul style="list-style-type: none"> <li>Log String.</li> </ul>

##### Returns

none

## 9.9.4 Enumeration Type Documentation

## 9.9.4.1 enum litefw\_fileimgtype

Enumerator

*eFILE\_TYPE\_NONE*  
*eFILE\_TYPE\_CAR\_PRI*  
*eFILE\_TYPE\_OEM\_PRI*  
*eFILE\_TYPE\_COMPO\_PRI*

## 9.9.4.2 enum litefw\_Fw\_Type

Enumerator

*eFW\_TYPE\_MBN\_GOBI*  
*eFW\_TYPE\_MBN*  
*eFW\_TYPE\_CWE*  
*eFW\_TYPE\_NVU*  
*eFW\_TYPE\_SPK*  
*eFW\_TYPE\_INVALID*  
*eFW\_TYPE\_CWE\_NVU*

## 9.9.4.3 enum litefw\_fwdwl\_error\_codes

Enumerator

*eSDP\_FWDWL\_SUCCESS*  
*eSDP\_FWDWL\_ERR\_GENERAL*  
*eSDP\_FWDWL\_ERR\_SDK*  
*eSDP\_FWDWL\_ERR\_SET\_CBK*  
*eSDP\_FWDWL\_ERR\_PATH\_TOO\_LONG*  
*eSDP\_FWDWL\_ERR\_PATH\_NOT\_SPECIFIED*  
*eSDP\_FWDWL\_ERR\_FW\_UPGRADE*  
*eSDP\_FWDWL\_ERR\_INVALID\_DEV*  
*eSDP\_FWDWL\_ERR\_INVALID\_PATH*  
*eSDP\_FWDWL\_ERR\_TIMEOUT*  
*eSDP\_FWDWL\_ERR\_FAIL*  
*eSDP\_FWDWL\_ERR\_PRI\_FAIL*  
*eSDP\_FWDWL\_ERR\_FW\_VERSION\_FAIL*  
*eSDP\_FWDWL\_ERR\_SDP\_TIMEOUT*  
*eFIREHOSE\_ERR\_SECBOOT\_INVALID\_CERT\_CHAIN*  
*eSDP\_FWDWL\_ERR\_END*

#### 9.9.4.4 enum litefw\_imagetype

Enumerator

***eIMAGE\_TYPE\_MIN***  
***eIMAGE\_TYPE\_BOOT***  
***eIMAGE\_TYPE\_APPL***  
***eIMAGE\_TYPE\_APPS***  
***eIMAGE\_TYPE\_FILE***  
***eIMAGE\_TYPE\_SPKG***  
***eIMAGE\_TYPE\_MODM***  
***eIMAGE\_TYPE\_USER***  
***eIMAGE\_TYPE\_KEYS***  
***eIMAGE\_TYPE\_MAX***  
***eIMAGE\_TYPE\_INVALID***  
***eIMAGE\_TYPE\_ANY***

#### 9.9.4.5 enum litefw\_Models

Enumerator

***eModel\_Unknown***  
***eModel\_9X15***  
***eModel\_WP9X15***  
***eModel\_9X30***  
***eModel\_9x07***  
***eModel\_9x06***

#### 9.9.4.6 enum litefw\_QDL\_FLOW\_CONTROLS

Enumerator

***eQDL\_HW\_FLOW\_Unknown***  
***eQDL\_HW\_FLOW\_INIT***  
***eQDL\_HW\_FLOW\_ENABLE***  
***eQDL\_HW\_FLOW\_DISABLE***

#### 9.9.4.7 enum litefw\_QDL\_MODEs

Enumerator

***eQDL\_MODE\_Unknown***  
***eQDL\_MODE\_INIT***  
***eQDL\_MODE\_TTYUSB***  
***eQDL\_MODE\_UART0***

### 9.9.5 Function Documentation

**9.9.5.1** `int litefw_BuildImagesPreferenceRequest ( litefw_FirmwareInfo info, pack_fms_SetImagesPreference_t * pack )`

This API Build Image Preference Request Using Firmware Information.

#### Parameters

in	<i>info</i>	<ul style="list-style-type: none"> <li>See <a href="#">litefw_FirmwareInfo</a> for more information.</li> </ul>
out	<i>pack</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_fms_SetImagesPreference_t</a> for more information.</li> </ul>

#### Returns

0 on success, `litefw_fwdwl_error_codes` error value otherwise

#### See Also

See [litefw\\_fwdwl\\_error\\_codes](#) for error values

**9.9.5.2** `int litefw_CalculateImageMask ( unpack_fms_SetImagesPreference_t SetPrefRspFromModem )`

This API Calculate Image Mask for Firmware Download.

#### Parameters

in	<i>SetPrefRsp-FromModem</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_fms_SetImagesPreference_t</a> for more information.</li> </ul>
----	-----------------------------	--

#### Returns

Image Mask

- `IMG_MASK_MDM` | `IMG_MASK_PRI`

**9.9.5.3** `int litefw_CheckValidFirmwareInfo ( litefw_FirmwareInfo info )`

This API Check Valid Firmware Information to build Image Preference Request.

#### Parameters

in	<i>info</i>	<ul style="list-style-type: none"> <li>See <a href="#">litefw_FirmwareInfo</a> for more information.</li> </ul>
----	-------------	---

#### Returns

0 on success, -1 error value otherwise

**9.9.5.4** unsigned int `litefw_DownloadFW` ( char \* *plmagePath*, char \* *szTTYPath*, int *iFWImageType*, int *image\_mask*, int *iModelFamily* )

This API Download Firmware.

#### Parameters

in	<i>plmagePath</i>	<ul style="list-style-type: none"> <li>Firmware Folder Path.</li> </ul>
in	<i>szTTYPath</i>	<ul style="list-style-type: none"> <li>QDL Device Path.</li> </ul>
in	<i>iFWImageType</i>	<ul style="list-style-type: none"> <li>Firmware Type.</li> <li>See <a href="#">litefw_Fw_Type</a></li> </ul>
in	<i>image_mask</i>	<ul style="list-style-type: none"> <li>Image Mask. <ul style="list-style-type: none"> <li>– IMG_MASK_MDM   IMG_MASK_PRI</li> </ul> </li> </ul>
in	<i>iModelFamily</i>	<ul style="list-style-type: none"> <li>Modem Family.</li> <li>See <a href="#">litefw_Models</a></li> </ul>

#### Returns

0 on success, `litefw_fwdwl_error_codes` error value otherwise

#### See Also

See [litefw\\_fwdwl\\_error\\_codes](#) for error values

**9.9.5.5** int `litefw_ExtractFirmwareParametersByPath` ( char \* *plmagePath*, `litefw_FirmwareInfo` \* *info* )

This API Extrac Firmwre Parameters From Path.

#### Parameters

in	<i>plmagePath</i>	<ul style="list-style-type: none"> <li>Firmware Folder Path.</li> </ul>
out	<i>info</i>	<ul style="list-style-type: none"> <li>See <a href="#">litefw_FirmwareInfo</a> for more information.</li> </ul>

#### Returns

0 on success, `litefw_fwdwl_error_codes` error value otherwise

#### See Also

See [litefw\\_fwdwl\\_error\\_codes](#) for error values

#### 9.9.5.6 int `litefw_ExtractFirmwarePartNoByPath` ( char \* *plmagePath*, `litefw_FirmwarePartNo` \* *partno* )

This API Extrace Firmware Part Number From Path.

##### Parameters

in	<i>plmagePath</i>	<ul style="list-style-type: none"><li>Firmware Folder Path.</li></ul>
out	<i>partno</i>	<ul style="list-style-type: none"><li>See <a href="#">litefw_FirmwarePartNo</a> for more information.</li></ul>

##### Returns

0 on success, `litefw_fwdwl_error_codes` error value otherwise

##### See Also

See [litefw\\_fwdwl\\_error\\_codes](#) for error values

#### 9.9.5.7 int `litefw_getFileType` ( char \* *szPath* )

This API Get File Type By Path

##### Parameters

in	<i>szPath</i>	<ul style="list-style-type: none"><li>See <a href="#">litefw_FirmwareInfo</a> for more information.</li></ul>
----	---------------	---

##### Returns

`eFW_TYPE_INVALID` on error, `litefw_Fw_Type` value otherwise

##### See Also

See [litefw\\_Fw\\_Type](#) for values

#### 9.9.5.8 int `litefw_GetModelFamily` ( char \* *pModelString* )

This API Get Model Famliy from a model string.

##### Parameters

in	<i>pModelString</i>	<ul style="list-style-type: none"><li>Model String.</li></ul>
----	---------------------	---

##### Returns

`eModel_Unknown` on Error, `litefw_Models` value otherwise

**See Also**

See [litefw\\_Models](#) for values

**9.9.5.9 int litefw\_GetQTLDownloadMode ( )**

This API Get QDL port donwload mode.

**Returns**

eQDL\_MODE\_Unknown on Error, litefw\_QDL\_MODEs value otherwise

**9.9.5.10 int litefw\_GetQTLHWFlowControl ( )**

This API Get QDL port Hardware flow control.

**Returns**

eQDL\_HW\_FLOW\_Unknown on Error, litefw\_QDL\_FLOW\_CONTROLS value otherwise

**9.9.5.11 char\* litefw\_GetVersion ( )**

This API Get Lib SDP Version.

**Returns**

Version String

**9.9.5.12 void litefw\_logsenable ( int *log\_en* )**

This function enable/disable lite firmware debug logs.

**Parameters**

<i>in</i>	<i>log_en</i>	<ul style="list-style-type: none"> <li>Logs enable/disable flag.</li> </ul>
-----------	---------------	---

**9.9.5.13 int litefw\_set\_log\_func ( litefwlogger \* *func* )**

This API Set Custom Log function.

**Parameters**

<i>in</i>	<i>func</i>	<ul style="list-style-type: none"> <li>See <a href="#">litefwlogger</a> for more information.</li> </ul>
-----------	-------------	--

**Returns**

none



9.9.5.14 int `litefw_SetQTLDownloadMode` ( int *iMode* )

This API set QDL port donwload mode.

## Parameters

in	<i>iMode</i>	<ul style="list-style-type: none"> <li>QDL Download Mode. <ul style="list-style-type: none"> <li>litefw_QDL_MODEs</li> </ul> </li> </ul>
----	--------------	--

## Returns

0 on success, -1 error value otherwise

9.9.5.15 int `litefw_SetQTLHWFlowControl` ( int *iMode* )

This API Set QDL port Hardware flow control.

## Parameters

in	<i>iMode</i>	<ul style="list-style-type: none"> <li>litefw_QDL_FLOW_CONTROLS</li> </ul>
----	--------------	--

## Returns

0 on success, -1 error value otherwise

9.9.5.16 void `litefw_SetReadBlockSize` ( unsigned long *IBlockSize* )

This API Set Read BlockSize.

## Parameters

in	<i>IBlockSize</i>	<ul style="list-style-type: none"> <li>Firmware Read Block Size.</li> </ul>
----	-------------------	---

## Returns

none

9.9.5.17 enum `eQCWWANError` `litefw_SLQSGetFirmwareFileInfo` ( char \* *fullFilePath*, `litefw_FirmwareFileInfo` \* *info*, uint8\_t \* *pNumOfItems* )

This API Get the firmware information of the provided image

## Parameters

in	<i>fullFilePath</i>	<ul style="list-style-type: none"> <li>file path or directory of the firmware image</li> </ul>
----	---------------------	--

out	<i>info</i>	<ul style="list-style-type: none"> <li>• See <a href="#">litefw_FirmwareFileInfo</a>, application should allocate the memory</li> <li>• for the struct array, this API will fill in the details in the provided storage</li> </ul>
out	<i>pNumOfItems</i>	<ul style="list-style-type: none"> <li>• number of <a href="#">litefw_FirmwareFileInfo</a></li> </ul>

**Returns**

enum eQCWWANError

#### 9.9.5.18 int [litefw\\_switch\\_9x07\\_to\\_downloadmode](#) ( char \* *szTTYPath* )

This API switch 9x07 modem to download mode.

**Parameters**

in	<i>szTTYPath</i>	<ul style="list-style-type: none"> <li>• QDL Device Path.</li> </ul>
----	------------------	--

**Returns**

0 on success, -1 error value otherwise

#### 9.9.5.19 int [litefw\\_switch\\_to\\_BootHoldMode](#) ( char \* *szTTYPath* )

This API switch modem to boot hold modem via QDL port.

**Parameters**

in	<i>szTTYPath</i>	<ul style="list-style-type: none"> <li>• QDL Device Path.</li> </ul>
----	------------------	--

**Returns**

0 on success, -1 error value otherwise

## 9.10 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](#)
- struct [unpack\\_loc\\_SetOperationMode\\_Ind\\_t](#)
- struct [unpack\\_loc\\_DeleteAssistData\\_Ind\\_t](#)
- struct [loc\\_satelliteInfo](#)
- struct [unpack\\_loc\\_GnssSvInfo\\_Ind\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectUTCTime\\_t](#)
- struct [altSrcInfo\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectPosition\\_t](#)
- struct [pack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#)
- struct [sensorData\\_t](#)
- struct [tempData\\_t](#)
- struct [pack\\_loc\\_SLQSLOCInjectSensorData\\_t](#)
- struct [unpack\\_loc\\_EventNMEA\\_Ind\\_t](#)
- struct [pack\\_loc\\_SLQSLOCGetServer\\_t](#)
- struct [loc\\_IPv4Info](#)
- struct [loc\\_IPv6Info](#)
- struct [loc\\_urlAddr](#)
- struct [unpack\\_loc\\_GetServer\\_Ind\\_t](#)
- struct [unpack\\_loc\\_CradleMountCallback\\_Ind\\_t](#)
- struct [unpack\\_loc\\_EventTimeSyncCallback\\_Ind\\_t](#)
- struct [unpack\\_loc\\_InjectTimeSyncDataCallback\\_Ind\\_t](#)
- struct [unpack\\_loc\\_InjectSensorDataCallback\\_Ind\\_t](#)
- struct [loc\\_accelAcceptReady](#)
- struct [loc\\_gyroAcceptReady](#)
- struct [loc\\_accelTempAcceptReady](#)
- struct [loc\\_gyroTempAcceptReady](#)
- struct [unpack\\_loc\\_SensorStreamingCallback\\_Ind\\_t](#)
- struct [unpack\\_loc\\_InjectUTCTimeCallback\\_Ind\\_t](#)
- struct [unpack\\_loc\\_InjectPositionCallback\\_Ind\\_t](#)
- struct [loc\\_IPv4Config](#)
- struct [loc\\_IPv6Config](#)
- struct [loc\\_URLAddrInfo](#)
- struct [pack\\_loc\\_SLQSLOCSetServer\\_t](#)

- struct [unpack\\_loc\\_SetServer\\_Ind\\_t](#)
- struct [unpack\\_loc\\_SLQSLOCGetOpMode\\_t](#)
- struct [unpack\\_loc\\_GetOpMode\\_Ind\\_t](#)
- struct [loc\\_FixCriteriaStatusTlv](#)
- struct [loc\\_HorAccuracyLvITlv](#)
- struct [loc\\_IntermediateRptStateTlv](#)
- struct [loc\\_MinIntervalTlv](#)
- struct [loc\\_AppProviderInfoTlv](#)
- struct [unpack\\_loc\\_FixCriteria\\_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 [LOCEVENTMASKSETSPSTREAMINGREPORT](#) 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
- #define [MAX\\_SENSOR\\_DATA\\_LEN](#) 64
- #define [MAX\\_TEMP\\_DATA\\_LEN](#) 64
- #define [MAX\\_LOC\\_NMEA\\_STR\\_LEN](#) 201

## Typedefs

- typedef [unpack\\_loc\\_Delete\\_Assist\\_Data\\_t](#) [unpack\\_loc\\_DeleteAssistData\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_SLQSLOCInjectUTCTime\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_SLQSLOCInjectPosition\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_SLQSLOCInjectSensorData\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_SLQSLOCGetServer\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_loc\\_GetFixCriteria\\_t](#)

## 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)
- int [unpack\\_loc\\_SetOperationMode\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetOperationMode\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_DeleteAssistData\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_DeleteAssistData\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_GnssSvInfo\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_GnssSvInfo\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCInjectUTCTime](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectUTCTime\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectUTCTime](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCInjectUTCTime\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCInjectPosition](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectPosition\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectPosition](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCInjectPosition\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCSetCradleMountConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#) \*reqArg)

- int [unpack\\_loc\\_SLQSLOCSetCradleMountConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCSetCradleMountConfig\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCInjectSensorData](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCInjectSensorData\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCInjectSensorData](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCInjectSensorData\\_t](#) \*pOutput)
- int [unpack\\_loc\\_EventNMEA\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_EventNMEA\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCGetServer](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCGetServer\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCGetServer](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCGetServer\\_t](#) \*pOutput)
- int [unpack\\_loc\\_GetServer\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_GetServer\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_CradleMountCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_CradleMountCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_EventTimeSyncCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_EventTimeSyncCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_InjectTimeSyncDataCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_InjectTimeSyncDataCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_InjectSensorDataCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_InjectSensorDataCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_SensorStreamingCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SensorStreamingCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_InjectUTCTimeCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_InjectUTCTimeCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_loc\\_InjectPositionCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_InjectPositionCallback\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCSetServer](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_loc\\_SLQSLOCSetServer\\_t](#) \*reqArg)
- int [unpack\\_loc\\_SLQSLOCSetServer](#) (uint8\_t \*pResp, uint16\_t respLen)
- int [unpack\\_loc\\_SetServer\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SetServer\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_SLQSLOCGetOpMode](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_loc\\_SLQSLOCGetOpMode](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_SLQSLOCGetOpMode\\_t](#) \*pOutput)
- int [unpack\\_loc\\_GetOpMode\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_GetOpMode\\_Ind\\_t](#) \*pOutput)
- int [pack\\_loc\\_GetFixCriteria](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, void \*reqArg)
- int [unpack\\_loc\\_GetFixCriteria](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_GetFixCriteria\\_t](#) \*pOutput)
- int [unpack\\_loc\\_FixCriteria\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_loc\\_FixCriteria\\_Ind\\_t](#) \*pOutput)

## 9.10.1 Macro Definition Documentation

### 9.10.1.1 #define LOC\_UINT8\_MAX\_STRING\_SZ 255

### 9.10.1.2 #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.

### 9.10.1.3 #define LOCEVENTMASKENGINESTATE 0x00000080

The control point must enable this mask to receive engine state report event indications.

**9.10.1.4 #define LOCEVENTMASKFIXSESSIONSTATE 0x00000100**

The control point must enable this mask to receive fix session status report event indications.

**9.10.1.5 #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.

**9.10.1.6 #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.

**9.10.1.7 #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.

**9.10.1.8 #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).

**9.10.1.9 #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.

**9.10.1.10 #define LOCEVENTMASKINJECTPOSITIONREQ 0x00000040**

The control point must enable this mask to receive position injection request event indications.

**9.10.1.11 #define LOCEVENTMASKINJECTPREDICTEDORBITSREQ 0x00000020**

The control point must enable this mask to receive predicted orbits request event indications.

**9.10.1.12 #define LOCEVENTMASKINJECTTIMERREQ 0x00000010**

The control point must enable this mask to receive time injection request event indications.

**9.10.1.13 #define LOCEVENTMASKINJECTWIFIAPDATAREQ 0x00200000**

The control point must enable this mask to receive Wi-Fi Access Point (AP) data inject request event indications.

**9.10.1.14 #define LOCEVENTMASKINVALIDVALUE 0xFFFFFFFF**

Invalid Event Mask

**9.10.1.15 #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.

**9.10.1.16 #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.

**9.10.1.17 #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.

**9.10.1.18 #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.

**9.10.1.19 #define LOCEVENTMASKNINOTIFYVERIFYREQ 0x00000008**

The control point must enable this mask to receive NI Notify/Verify request event indications.

**9.10.1.20 #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.

**9.10.1.21 #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.

**9.10.1.22 #define LOCEVENTMASKPOSITIONREPORT 0x00000001**

The control point must enable this mask to receive position report event indications.

**9.10.1.23 #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.).

**9.10.1.24 #define LOCEVENTMASKSETSPISTREAMINGREPORT 0x00001000**

The control point must enable this mask to receive Stationary Position Indicator (SPI) streaming report indications.



9.10.1.25 `#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.

9.10.1.26 `#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.).

9.10.1.27 `#define LOCEVENTMASKWIFIREQ 0x00000200`

The control point must enable this mask to receive Wi-Fi position request event indications.

9.10.1.28 `#define MAX_LOC_NMEA_STR_LEN 201`9.10.1.29 `#define MAX_SENSOR_DATA_LEN 64`9.10.1.30 `#define MAX_TEMP_DATA_LEN 64`

## 9.10.2 Typedef Documentation

9.10.2.1 `typedef unpack_loc_Delete_Assist_Data_t unpack_loc_DeleteAssistData_t`9.10.2.2 `typedef unpack_result_t unpack_loc_GetFixCriteria_t`9.10.2.3 `typedef unpack_result_t unpack_loc_SLQSLOCGetServer_t`9.10.2.4 `typedef unpack_result_t unpack_loc_SLQSLOCInjectPosition_t`9.10.2.5 `typedef unpack_result_t unpack_loc_SLQSLOCInjectSensorData_t`9.10.2.6 `typedef unpack_result_t unpack_loc_SLQSLOCInjectUTCTime_t`9.10.2.7 `typedef unpack_result_t unpack_loc_SLQSLOCSetCradleMountConfig_t`

## 9.10.3 Enumeration Type Documentation

## 9.10.3.1 anonymous enum

Enumerator

***eQMI\_LOC\_SESS\_STATUS\_SUCCESS***

***eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS***

***eQMI\_LOC\_SESS\_STATUS\_FAILURE***

***eQMI\_LOC\_SESS\_STATUS\_TIMEOUT***

## 9.10.4 Function Documentation

9.10.4.1 `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, 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

9.10.4.2 int pack\_loc\_EventRegister ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_loc\_EventRegister\_t \* *reqArg* )

Event Register pack

**Parameters**

in, 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

9.10.4.3 int pack\_loc\_GetFixCriteria ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, void \* *reqArg* )

Pack Get Fix Criteria.

**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

9.10.4.4 `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, 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

9.10.4.5 `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, 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

9.10.4.6 `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, 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

9.10.4.7 `int pack_loc_SLQSLOCGetOpMode ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Pack Get Operation mode.

**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

9.10.4.8 `int pack_loc_SLQSLOCGetServer ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCGetServer_t * reqArg )`

Pack get server.

**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

**Note**

The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through [unpack\\_loc\\_GetServer\\_Ind](#) callback. If successful, the callback also contains the A-GPS server address.

**See Also**

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

9.10.4.9 `int pack_loc_SLQSLOCInjectPosition ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCInjectPosition_t * reqArg )`

Pack inject position.

#### 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

9.10.4.10 `int pack_loc_SLQSLOCInjectSensorData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCInjectSensorData_t * reqArg )`

Pack inject sensor data.

#### 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

9.10.4.11 `int pack_loc_SLQSLOCInjectUTCtime ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCInjectUTCtime_t * reqArg )`

Pack inject UTC time.

#### 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

9.10.4.12 `int pack_loc_SLQSLOCSetCradleMountConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCSetCradleMountConfig_t * reqArg )`

Pack set cradle mount configure.

**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

9.10.4.13 `int pack_loc_SLQSLOCSetServer ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_SLQSLOCSetServer_t * reqArg )`

Pack Set server.

**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

**Note**

If multiple types of addresses are specified in the request, the IPv4 address takes precedence over the IPv6 address and the IPv6 address takes precedence over the URL address. The request is acknowledged through the response, but the SUCCESS/FAILURE status is sent through [unpack\\_loc\\_SetServer\\_Ind](#) callback

**See Also**

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

9.10.4.14 `int pack_loc_Start ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Start_t * reqArg )`

LOC Start 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

9.10.4.15 `int pack_loc_Stop ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_loc_Stop_t * reqArg )`

Loc Stop 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

9.10.4.16 `int unpack_loc_BestAvailPos_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_BestAvailPos_Ind_t * pOutput )`

Loc Best Avial 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

9.10.4.17 `int unpack_loc_CradleMountCallback_Ind ( uint8_t * pResp, uint16_t respLen,  
unpack_loc_CradleMountCallback_Ind_t * pOutput )`

Unpack the Cradle Mount Indication.

## 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

9.10.4.18 `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

9.10.4.19 `int unpack_loc_DeleteAssistData_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_DeleteAssistData_Ind-  
_t * pOutput )`

Unpack the status of delete the location engine assistance data

## 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

9.10.4.20 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

9.10.4.21 int unpack\_loc\_EventNMEA\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_EventNMEA\_Ind\_t \* *pOutput* )

Unpack the NMEA Event Indication.

## 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

9.10.4.22 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

**9.10.4.23** int unpack\_loc\_EventTimeSyncCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_EventTimeSyncCallback\_Ind\_t \* *pOutput* )

Unpack the Event Time Sync indication.

**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

**9.10.4.24** int unpack\_loc\_FixCriteria\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_FixCriteria\_Ind\_t \* *pOutput* )

Unpack the Get Fix Criteria Indication.

**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

**9.10.4.25** int unpack\_loc\_GetFixCriteria ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_GetFixCriteria\_t \* *pOutput* )

Get Fix Criteria 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

9.10.4.26 `int unpack_loc_GetOpMode_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_GetOpMode_Ind_t * pOutput )`

Unpack the Set Server Indication.

## 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

9.10.4.27 `int unpack_loc_GetServer_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_GetServer_Ind_t * pOutput )`

Unpack the Get Server Indication.

## 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

9.10.4.28 `int unpack_loc_GnssSvInfo_Ind ( uint8_t * pResp, uint16_t respLen, unpack_loc_GnssSvInfo_Ind_t * pOutput )`

Unpack the GNSS SV Info Indication.

## 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

**9.10.4.29** int unpack\_loc\_InjectPositionCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_InjectPositionCallback\_Ind\_t \* *pOutput* )

Unpack the Inject position indication.

**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

**9.10.4.30** int unpack\_loc\_InjectSensorDataCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_InjectSensorDataCallback\_Ind\_t \* *pOutput* )

Unpack the inject sensor data indication.

**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

**9.10.4.31** int unpack\_loc\_InjectTimeSyncDataCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t \* *pOutput* )

Unpack the inject time sync data indication.

**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

**9.10.4.32** int unpack\_loc\_InjectUTCtimeCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_InjectUTCtimeCallback\_Ind\_t \* *pOutput* )

Unpack the Inject UTC time indication.

## 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

**9.10.4.33** 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

**9.10.4.34** int unpack\_loc\_SensorStreamingCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_loc\_SensorStreamingCallback\_Ind\_t \* *pOutput* )

Unpack the Sensor streaming status indication.

## 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

9.10.4.35 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

9.10.4.36 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

9.10.4.37 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

9.10.4.38 int unpack\_loc\_SetOperationMode\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SetOperationMode\_Ind\_t \* *pOutput* )

Unpack the engine to use the specified operation mode.

## 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

9.10.4.39 int unpack\_loc\_SetServer\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SetServer\_Ind\_t \* *pOutput* )

Unpack the Set Server Indication.

## 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

9.10.4.40 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

**9.10.4.41** int unpack\_loc\_SLQSLOCGetOpMode ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SLQSLOCGetOpMode\_t \* *pOutput* )

Get 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

**9.10.4.42** int unpack\_loc\_SLQSLOCGetServer ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SLQSLOCGetServer\_t \* *pOutput* )

Unpack get server.

**Parameters**

in	<i>pResp</i>	qmi response
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

**9.10.4.43** int unpack\_loc\_SLQSLOCInjectPosition ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_loc\_SLQSLOCInjectPosition\_t \* *pOutput* )

Unpack inject position.

**Parameters**

in	<i>pResp</i>	qmi response
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

9.10.4.44 `int unpack_loc_SLQSLOCInjectSensorData ( uint8_t * pResp, uint16_t respLen, unpack_loc_SLQSLOCInjectSensorData_t * pOutput )`

Unpack inject sensor data.

## Parameters

in	<i>pResp</i>	qmi response
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

9.10.4.45 `int unpack_loc_SLQSLOCInjectUTCTime ( uint8_t * pResp, uint16_t respLen, unpack_loc_SLQSLOCInjectUTCTime_t * pOutput )`

Unpack inject UTC time.

## Parameters

in	<i>pResp</i>	qmi response
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

9.10.4.46 `int unpack_loc_SLQSLOCSetCradleMountConfig ( uint8_t * pResp, uint16_t respLen, unpack_loc_SLQSLOCSetCradleMountConfig_t * pOutput )`

Unpack set cradle mount configure.

## Parameters

in	<i>pResp</i>	qmi response
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

#### 9.10.4.47 int unpack\_loc\_SLQSLOCSetServer ( uint8\_t \* *pResp*, uint16\_t *respLen* )

Unpack set server.

**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

#### 9.10.4.48 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

#### 9.10.4.49 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

## 9.11 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 [nas\\_lteBandPrefExt](#)
- struct [nas\\_ciotAcqOrderPref](#)
- struct [nas\\_nr5gBandPref](#)
- struct [pack\\_nas\\_SLQSSetSysSelectionPrefExt\\_t](#)
- struct [pack\\_nas\\_SLQSSetBandPreference\\_t](#)
- struct [pack\\_nas\\_SLQSNasIndicationRegisterExt\\_t](#)
- struct [pack\\_nas\\_SLQSNasIndicationRegisterV2\\_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\\_homeNwMNC3GppTlv](#)
- struct [nas\\_nwNameSrc3GppTlv](#)
- struct [unpack\\_nas\\_SLQSGetHomeNetwork\\_t](#)
- struct [nas\\_SrvStatusInfo](#)
- struct [nas\\_GSMSrvStatusInfo](#)
- struct [nas\\_sysInfoCommon](#)
- struct [nas\\_CDMASysInfo](#)
- struct [nas\\_HDRSysInfo](#)
- struct [nas\\_GSMSSysInfo](#)
- struct [nas\\_WCDMASysInfo](#)
- struct [nas\\_LTESysInfo](#)
- struct [nas\\_AddCDMASysInfo](#)
- struct [nas\\_AddSysInfo](#)
- struct [nas\\_CallBarringSysInfo](#)
- struct [nas\\_LteCiotOpModeTlv](#)
- struct [nas\\_NR5GSerStatTlv](#)
- struct [nas\\_NR5GSystemInfoTlv](#)
- struct [nas\\_NR5GCellStatusTlv](#)
- struct [unpack\\_nas\\_SLQSGetSysInfo\\_t](#)
- struct [nas\\_LteEmbmsCoverageTlv](#)

- struct [nas\\_SimRejInfoTlv](#)
- struct [nas\\_lmsVoiceSupportLteTlv](#)
- struct [nas\\_LteVoiceDomainTlv](#)
- struct [nas\\_SrvRegRestrictionTlv](#)
- struct [nas\\_LteRegDomainTlv](#)
- struct [nas\\_LteEmbmsTraceIdTlv](#)
- struct [nas\\_NR5GSrvStatusTlv](#)
- struct [nas\\_NR5GSysInfoTlv](#)
- struct [nas\\_NR5GCellStatusInfoTlv](#)
- struct [unpack\\_nas\\_SLQSGetSysInfoV2\\_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 [nas\\_QmisNasSlqsNasPCICellInfo](#)
- struct [nas\\_QmisNasSlqsNasPCIInfo](#)
- struct [nas\\_lteOpModeTlv](#)
- struct [unpack\\_nas\\_PerformNetworkScan\\_t](#)
- struct [nas\\_networkNameSrcTlv](#)
- struct [unpack\\_nas\\_SLQSPerformNetworkScanV2\\_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\\_CDMA SysInfoExt](#)
- struct [nas\\_callBarStatus](#)
- struct [unpack\\_nas\\_SLQSGetServingSystem\\_t](#)
- struct [nas\\_MNCPCSDigitStatus](#)
- struct [unpack\\_nas\\_SLQSGetServingSystemV2\\_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 [unpack\\_nas\\_GetANAAAAuthenticationStatus\\_t](#)
- struct [unpack\\_nas\\_GetACCOLC\\_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\\_TDSCDMARSCPThresh](#)
- 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 [NASPhyCaAggScellArray](#)
- 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 [NASAcqOrderPrefTlv](#)
- struct [NASRatDisabledMaskTlv](#)
- struct [NASCiotLteOpModePrefTlv](#)
- struct [NASLteM1BandPrefTlv](#)
- struct [NASLteNB1BandPrefTlv](#)
- struct [NASCiotAcqOrderPrefTlv](#)
- struct [NASNr5gBandPrefTlv](#)
- 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\\_SLQSNasSwiIndicationRegister\\_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\\_WCDMACellInfoExt](#)
- struct [nas\\_umtsInstArr](#)
- struct [nas\\_geranInstArr](#)
- struct [nas\\_UMTSExtInfo](#)
- struct [nas\\_LteEarfcnInfo](#)
- struct [unpack\\_nas\\_SLQSNasGetCellLocationInfoV2\\_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 [nas\\_PhyCaAggDIBW](#)
- struct [nas\\_NumScellsConfig](#)
- struct [unpack\\_nas\\_SetNasLTECphyCaIndCallback\\_ind\\_t](#)
- struct [nas\\_RxSigInfo](#)
- struct [nas\\_SccRxInfo](#)
- struct [unpack\\_nas\\_SLQSSwiGetLteSccRxInfo\\_t](#)
- struct [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t](#)
- struct [pack\\_nas\\_InitiateDomainAttach\\_t](#)
- struct [pack\\_nas\\_SetCDMANetworkParameters\\_t](#)
- struct [unpack\\_nas\\_SLQSNasGetHDRColorCode\\_t](#)
- struct [pack\\_nas\\_SLQSNasGetTxRxInfo\\_t](#)
- struct [nas\\_rxInfo](#)
- struct [nas\\_txInfo](#)

- struct [unpack\\_nas\\_SLQSNasGetTxRxInfo\\_t](#)
- struct [nas\\_OperatorPLMNData](#)
- struct [nas\\_operatorPLMNList](#)
- struct [nas\\_serviceProviderName](#)
- struct [nas\\_PLMNNetworkNameData](#)
- struct [nas\\_PLMNNetworkName](#)
- struct [nas\\_operatorNameString](#)
- struct [unpack\\_nas\\_SLQSGetOperatorNameData\\_t](#)
- struct [pack\\_nas\\_SLQSNasGet3GPP2Subscription\\_t](#)
- struct [nas\\_namName](#)
- struct [nas\\_dirNum](#)
- struct [nas\\_sidNid](#)
- struct [nas\\_homeSIDNID](#)
- struct [nas\\_minBasedIMSI](#)
- struct [nas\\_trueIMSI](#)
- struct [nas\\_CDMAChannel](#)
- struct [nas\\_Mdn](#)
- struct [unpack\\_nas\\_SLQSNasGet3GPP2Subscription\\_t](#)
- struct [nas\\_protocolSubtypeElement](#)
- struct [unpack\\_nas\\_SLQSSwiGetHDRPersonality\\_t](#)
- struct [unpack\\_nas\\_SLQSSwiGetHDRProtSubtype\\_t](#)
- struct [pack\\_nas\\_SLQSSwiPSDetach\\_t](#)
- struct [unpack\\_nas\\_SLQSGetErrorRate\\_t](#)
- struct [nas\\_DRCParams](#)
- struct [nas\\_PilotSetParams](#)
- struct [nas\\_PilotSetData](#)
- struct [unpack\\_nas\\_SLQSSwiGetHRPDStats\\_t](#)
- struct [nas\\_ActPilotPNElement](#)
- struct [nas\\_NetworkStat1x](#)
- struct [nas\\_NetworkStatEVDO](#)
- struct [nas\\_DeviceConfigDetail](#)
- struct [nas\\_DataStatusDetail](#)
- struct [unpack\\_nas\\_SLQSSwiNetworkDebug\\_t](#)
- struct [nas\\_RSSIThresh](#)
- struct [nas\\_ECIOThresh](#)
- struct [nas\\_HDRSINRThresh](#)
- struct [nas\\_LTESNRThresh](#)
- struct [nas\\_IOTThresh](#)
- struct [nas\\_RSRQThresh](#)
- struct [nas\\_RSRPThresh](#)
- struct [nas\\_LTESigRptCfg](#)
- struct [nas\\_TDSCDMASINRCONFTThresh](#)
- struct [pack\\_nas\\_SLQSConfigSigInfo\\_t](#)
- struct [unpack\\_nas\\_GetHomeNetwork3GPP2\\_t](#)
- struct [nas\\_wcdmaUARFCN](#)
- struct [nas\\_lteEARFCN](#)
- struct [nas\\_ltePCI](#)
- struct [pack\\_nas\\_SLQSNASSwiSetChannelLock\\_t](#)
- struct [unpack\\_nas\\_SLQSNASSwiGetChannelLock\\_t](#)
- struct [unpack\\_nas\\_SLQSNASGeteDRXParams\\_t](#)
- struct [pack\\_nas\\_SLQSNASGeteDRXParamsExt\\_t](#)
- struct [unpack\\_nas\\_SLQSNASGeteDRXParamsExt\\_t](#)
- struct [pack\\_nas\\_SLQSNASSeteDRXParams\\_t](#)
- struct [nas\\_HDRPersonality\\_Ind\\_Data](#)
- struct [unpack\\_nas\\_SLQSSwiHDRPersonalityCallback\\_Ind\\_t](#)

- struct [nas\\_RankIndicatorTlv](#)
- struct [unpack\\_nas\\_SLQSSwiRandIndicatorCallback\\_Ind\\_t](#)
- struct [nas\\_RFBandInfoElements](#)
- struct [nas\\_RfDedicatedBandInfoElements](#)
- struct [nas\\_RfBandInfoExtFormatElements](#)
- struct [nas\\_RfBandwidthInfoElements](#)
- struct [nas\\_RfBandInfoList](#)
- struct [nas\\_RfDedicatedBandInfo](#)
- struct [nas\\_RfBandInfoExtFormat](#)
- struct [nas\\_RfBandwidthInfo](#)
- struct [nas\\_LTEOperationMode](#)
- struct [unpack\\_nas\\_SLQSNasGetRFInfo\\_t](#)
- struct [nas\\_EmerModeTlv](#)
- struct [nas\\_ModePrefTlv](#)
- struct [nas\\_BandPrefTlv](#)
- struct [nas\\_PRLPrefTlv](#)
- struct [nas\\_RoamPrefTlv](#)
- struct [nas\\_LTEBandPrefTlv](#)
- struct [nas\\_NetSelPrefTlv](#)
- struct [nas\\_SrvDomainPrefTlv](#)
- struct [nas\\_GWAcqOrderPrefTlv](#)
- struct [nas\\_AcqOrderPrefTlv](#)
- struct [nas\\_RatDisabledMaskTlv](#)
- struct [nas\\_CiotLteOpModePrefTlv](#)
- struct [nas\\_LteM1BandPrefTlv](#)
- struct [nas\\_LteNb1BandPrefTlv](#)
- struct [nas\\_CiotAcqOrderPrefTlv](#)
- struct [nas\\_BandPrefInfoTlv](#)
- struct [unpack\\_nas\\_SLQSGetSysSelectionPrefExt\\_t](#)
- struct [nas\\_TDSCDMABandPrefTlv](#)
- struct [nas\\_SrvRegRestricTlv](#)
- struct [nas\\_UsageSettingTlv](#)
- struct [nas\\_VoiceDomainPrefTlv](#)
- struct [unpack\\_nas\\_SLQSGetSysSelectionPrefExtV2\\_t](#)
- struct [pack\\_nas\\_PerformNetworkScanPCI\\_t](#)
- struct [nas\\_EdrxEnableType](#)
- struct [nas\\_EdrxCycleLength](#)
- struct [nas\\_EdrxPagingTimeWindow](#)
- struct [nas\\_EdrxRatType](#)
- struct [nas\\_EdrxCiotLteMode](#)
- struct [unpack\\_nas\\_SLQSNasEdrxChangeInfoCallBack\\_Ind\\_t](#)
- struct [nas\\_CsgId](#)
- struct [nas\\_PlmnID](#)
- struct [nas\\_LteOpMode](#)
- struct [unpack\\_nas\\_SLQSNasNetworkRejectCallback\\_Ind\\_t](#)
- struct [nas\\_ForbiddenNetworks3GPP](#)
- struct [unpack\\_nas\\_SLQSNASGetForbiddenNetworks\\_t](#)
- struct [nas\\_RFBandInfoTlv](#)
- struct [nas\\_RFDedicatedBandInfoTlv](#)
- struct [nas\\_RFBandInfoExtTlv](#)
- struct [nas\\_RFBandwidthInfoTlv](#)
- struct [nas\\_LTEOperationalModeTlv](#)
- struct [unpack\\_nas\\_SLQSNasRFBandInfoCallback\\_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_MAX_SCC_RX_INFO_INSTANCES` 255
- `#define NAS_NAM_NAME_LENGTH` 12
- `#define NAS_IMSI_M_S1_LENGTH` 7
- `#define NAS_IMSI_M_S2_LENGTH` 3
- `#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST` 255
- `#define NAS_MAX_SLQS_NAS_PCI_INFO_PLMN_LENGTH` 255
- `#define NAS_MAX_SLQS_NAS_PCI_INFO_LENGTH` 255
- `#define NAS_MCC_MNC_INST_LENGTH` 255
- `#define LITE_NW_SCAN_LTE_OP_MODE_MAX_LENGTH` 255
- `#define NAS_MAX_PHY_CA_AGG_SELL_ARRRY_SIZE` 255
- `#define LITE_MAX_NAS_3GPP2_MDN_LEN` 15
- `#define LITE_MAX_SLQS_NAS_NW_NAME_SRC_LENGTH` 255

## Typedefs

- `typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPref_t`
- `typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPrefExt_t`
- `typedef unpack_result_t unpack_nas_SLQSSetBandPreference_t`
- `typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterExt_t`
- `typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterV2_t`
- `typedef unpack_result_t unpack_nas_SLQSSetSignalStrengthsCallback_t`
- `typedef unpack_result_t unpack_nas_SetRFInfoCallback_t`
- `typedef unpack_result_t unpack_nas_SetLURejectCallback_t`
- `typedef unpack_result_t unpack_nas_SetACCOLC_t`
- `typedef unpack_result_t unpack_nas_SLQSNasConfigSigInfo2_t`
- `typedef unpack_result_t unpack_nas_SLQSIInitiateNetworkRegistration_t`
- `typedef unpack_result_t unpack_nas_SLQSNasSwiIndicationRegister_t`
- `typedef unpack_result_t unpack_nas_InitiateDomainAttach_t`
- `typedef unpack_result_t unpack_nas_SetCDMANetworkParameters_t`
- `typedef unpack_result_t unpack_nas_SLQSSwiPSDetach_t`
- `typedef unpack_result_t unpack_nas_SLQSConfigSigInfo_t`
- `typedef unpack_result_t unpack_nas_SLQSNASSwiSetChannelLock_t`
- `typedef unpack_result_t unpack_nas_SLQSNASSeteDRXParams_t`
- `typedef nas_BandPrefInfoTlv nas_NR5gBandPrefTlv`
- `typedef nas_BandPrefInfoTlv nas_LTEBandPrefExtTlv`

## Enumerations

- `enum LITEQMI_NAS_LTE_CPHY_SCELL_STATE {`  
`eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_DECONFIGURED =0x00,`  
`eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_DEACTIVATED =0x01,`  
`eLITEQMI_NAS_LTE_CPHY_SCELL_STATE_CONFIGURED_ACTIVATED =0x02 }`
- `enum LITEQMI_NAS_LTE_CPHY_CA_BW_NRB {`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_6 =0x00,`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_15 =0x01,`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_25 =0x02,`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_50 =0x03,`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_75 =0x04,`  
`eLITEQMI_NAS_LTE_CPHY_CA_BW_NRB_100 =0x05 }`

- enum LITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND {
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_1 = 120,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_2 = 121,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_3 = 122,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_4 = 123,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_5 = 124,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_6 = 125,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_7 = 126,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_8 = 127,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_9 = 128,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_10 = 129,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_11 = 130,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_12 = 131,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_13 = 132,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_14 = 133,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_17 = 134,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_33 = 135,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_34 = 136,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_35 = 137,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_36 = 138,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_37 = 139,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_38 = 140,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_39 = 141,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_40 = 142,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_18 = 143,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_19 = 144,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_20 = 145,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_21 = 146,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_24 = 147,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_25 = 148,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_41 = 149,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_42 = 150,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_43 = 151,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_23 = 152,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_26 = 153,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_32 = 154,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_125 = 155,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_126 = 156,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_127 = 157,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_28 = 158,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_29 = 159,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_30 = 160,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_66 = 161,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_250 = 162,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_46 = 163,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_27 = 164,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_31 = 165,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_71 = 166,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_47 = 167,
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_48 = 168 }
- 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_CELL_STATE_DECONFIGURED_LITE = 0x00,
eNAS_LTE_CPHY_CELL_STATE_CONFIGURED_DEACTIVATED_LITE = 0x01,
eNAS_LTE_CPHY_CELL_STATE_CONFIGURED_ACTIVATED_LITE = 0x02 }
```

- enum LITE\_TYPE\_OF\_SERVICE\_DOMAIN {
 LITE\_SYS\_SRV\_DOMAIN\_NO\_SRV = 0x00,
 LITE\_SYS\_SRV\_DOMAIN\_CS\_ONLY = 0x01,
 LITE\_SYS\_SRV\_DOMAIN\_PS\_ONLY = 0x02,
 LITE\_SYS\_SRV\_DOMAIN\_CS\_PS = 0x03,
 LITE\_SYS\_SRV\_DOMAIN\_CAMPED = 0x04 }

## 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\_SLQSSetSysSelectionPref (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int unpack\_nas\_SLQSSetSysSelectionPref (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSSetSysSelectionPref\_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, unpack\_nas\_SLQSSetSysSelectionPref\_t \*pOutput)
- int pack\_nas\_SLQSSetSysSelectionPrefExt (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, pack\_nas\_SLQSSetSysSelectionPrefExt\_t \*pReqParam)
- int unpack\_nas\_SLQSSetSysSelectionPrefExt (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSSetSysSelectionPrefExt\_t \*pOutput)
- int pack\_nas\_SLQSSetBandPreference (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, pack\_nas\_SLQSSetBandPreference\_t \*pReqParam)
- int unpack\_nas\_SLQSSetBandPreference (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSSetBandPreference\_t \*pOutput)
- 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, unpack\_nas\_SLQSNasIndicationRegisterExt\_t \*pOutput)
- int pack\_nas\_SLQSNasIndicationRegisterV2 (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, pack\_nas\_SLQSNasIndicationRegisterV2\_t \*pReqParam)
- int unpack\_nas\_SLQSNasIndicationRegisterV2 (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSNasIndicationRegisterV2\_t \*pOutput)
- 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\_valid\_nas\_SLQSNasGetSigInfo (uint8\_t \*pResp, uint8\_t u8Info)
- 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 unpack\_nas\_SLQSSetHomeNetwork (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSSetHomeNetwork\_t \*pOutput)
- int pack\_nas\_SLQSSetHomeNetwork (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int pack\_nas\_SLQSSetSysInfo (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int unpack\_nas\_SLQSSetSysInfo (uint8\_t \*pResp, uint16\_t respLen, unpack\_nas\_SLQSSetSysInfo\_t \*pOutput)

- int [unpack\\_nas\\_SLQSGetSysInfoV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetSysInfoV2\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetSysInfoV2](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- 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 [unpack\\_nas\\_SLQSPerformNetworkScanV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSPerformNetworkScanV2\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSPerformNetworkScanV2](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- 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 [unpack\\_nas\\_SLQSGetServingSystemV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetServingSystemV2\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetServingSystemV2](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_valid\\_nas\\_SLQSGetServingSystem](#) (uint8\_t \*pResp, uint8\_t u8Info)
- 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 [unpack\\_valid\\_nas\\_SLQSGetSignalStrength](#) (uint8\_t \*pResp, uint8\_t u8Info)
- 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, [unpack\\_nas\\_SLQSSetSignalStrengthsCallback\\_t](#) \*pOutput)
- 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, [unpack\\_nas\\_SetRFInfoCallback\\_t](#) \*pOutput)
- int [pack\\_nas\\_SetLURRejectCallback](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pBenable)
- int [unpack\\_nas\\_SetLURRejectCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetLURRejectCallback\\_t](#) \*pOutput)
- 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\\_GetCDMANetworkParameters\\_t](#) \*pOutput)
- int [unpack\\_valid\\_nas\\_GetCDMANetworkParameters](#) (uint8\_t \*pResp, uint8\_t u8Info)
- int [pack\\_nas\\_GetANAAAAuthenticationStatus](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_GetANAAAAuthenticationStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetANAAAAuthenticationStatus\\_t](#) \*pOutput)
- int [pack\\_nas\\_GetACCOLC](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)

- int [unpack\\_nas\\_GetACCOLC](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetACCOLC\\_t](#) \*pOutput)
- 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, [unpack\\_nas\\_SetACCOLC\\_t](#) \*pOutput)
- 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, [unpack\\_nas\\_SLQSNasConfigSigInfo2\\_t](#) \*pOutput)
- 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, [unpack\\_nas\\_SLQSInitiateNetworkRegistration\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasSwiIndicationRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasSwiIndicationRegister\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasSwiIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasSwiIndicationRegister\\_t](#) \*pOutput)
- 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 [unpack\\_nas\\_SLQSNasGetCellLocationInfoV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetCellLocationInfoV2\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasGetCellLocationInfoV2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- 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\\_SetNasLTECphyCaIndCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetNasLTECphyCaIndCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSSwiGetLteSccRxInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReq, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSSwiGetLteSccRxInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetLteSccRxInfo\\_t](#) \*pOutput)

- [int unpack\\_nas\\_SLQSNasTimerCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasTimerCallback\\_ind\\_t](#) \*pOutput)
- [int pack\\_nas\\_InitiateDomainAttach](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_InitiateDomainAttach\\_t](#) \*pReqParam)
- [int unpack\\_nas\\_InitiateDomainAttach](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_InitiateDomainAttach\\_t](#) \*pOutput)
- [int pack\\_nas\\_SetCDMANetworkParameters](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SetCDMANetworkParameters\\_t](#) \*pReqParam)
- [int unpack\\_nas\\_SetCDMANetworkParameters](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SetCDMANetworkParameters\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSNasGetHDRColorCode](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSNasGetHDRColorCode](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetHDRColorCode\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSNasGetTxRxInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasGetTxRxInfo\\_t](#) \*reqArg)
- [int unpack\\_nas\\_SLQSNasGetTxRxInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetTxRxInfo\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSGetOperatorNameData](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSGetOperatorNameData](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetOperatorNameData\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSNasGet3GPP2Subscription](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNasGet3GPP2Subscription\\_t](#) \*reqArg)
- [int unpack\\_nas\\_SLQSNasGet3GPP2Subscription](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGet3GPP2Subscription\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSSwiGetHDRPersonality](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSSwiGetHDRPersonality](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetHDRPersonality\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSSwiGetHDRProtSubtype](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSSwiGetHDRProtSubtype](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetHDRProtSubtype\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSSwiPSDetach](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSSwiPSDetach\\_t](#) \*pReqParam)
- [int unpack\\_nas\\_SLQSSwiPSDetach](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiPSDetach\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSGetErrorRate](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSGetErrorRate](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGetErrorRate\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSSwiGetHRPDStats](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSSwiGetHRPDStats](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiGetHRPDStats\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSSwiNetworkDebug](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_SLQSSwiNetworkDebug](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSSwiNetworkDebug\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSConfigSigInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSConfigSigInfo\\_t](#) \*pReqParam)
- [int unpack\\_nas\\_SLQSConfigSigInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSConfigSigInfo\\_t](#) \*pOutput)
- [int pack\\_nas\\_GetHomeNetwork3GPP2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_nas\\_GetHomeNetwork3GPP2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_GetHomeNetwork3GPP2\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSNASSwiSetChannelLock](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNASSwiSetChannelLock\\_t](#) \*pReqParam)
- [int unpack\\_nas\\_SLQSNASSwiSetChannelLock](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNASSwiSetChannelLock\\_t](#) \*pOutput)
- [int pack\\_nas\\_SLQSNASSwiGetChannelLock](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)



- int [unpack\\_nas\\_SLQSNASSwiGetChannelLock](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNAS-SwiGetChannelLock\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNASGeteDRXParams](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNASGeteDRXParams](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNAS-GeteDRXParams\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNASGeteDRXParamsExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNASGeteDRXParamsExt\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNASGeteDRXParamsExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNAS-GeteDRXParamsExt\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNASSeteDRXParams](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_SLQSNASSeteDRXParams\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNASSeteDRXParams](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNAS-SeteDRXParams\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSSwiHDRPersonalityCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SL-QSSwiHDRPersonalityCallback\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSSwiRandIndicatorCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQ-SSwiRandIndicatorCallback\\_Ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNasGetRFInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNasGetRFInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSNasGetRFInfo\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetSysSelectionPrefExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSGetSysSelectionPrefExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSGet-SysSelectionPrefExt\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSGetSysSelectionPrefExtV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQS-GetSysSelectionPrefExtV2\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSGetSysSelectionPrefExtV2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [pack\\_nas\\_PerformNetworkScanPCI](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_nas\\_-PerformNetworkScanPCI\\_t](#) \*pReqParam)
- int [unpack\\_nas\\_SLQSNasEdrxChangeInfoCallBack\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_S-LQSNasEdrxChangeInfoCallBack\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasNetworkRejectCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SL-QSNasNetworkRejectCallback\\_Ind\\_t](#) \*pOutput)
- int [pack\\_nas\\_SLQSNASGetForbiddenNetworks](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_nas\\_SLQSNASGetForbiddenNetworks](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQSN-ASGetForbiddenNetworks\\_t](#) \*pOutput)
- int [unpack\\_nas\\_SLQSNasRFBandInfoCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_nas\\_SLQS-NasRFBandInfoCallback\\_Ind\\_t](#) \*pOutput)

### 9.11.1 Macro Definition Documentation

- 9.11.1.1 `#define LITE_MAX_NAS_3GPP2_MDN_LEN 15`
- 9.11.1.2 `#define LITE_MAX_SLQS_NAS_NW_NAME_SRC_LENGTH 255`
- 9.11.1.3 `#define LITE_NW_SCAN_LTE_OP_MODE_MAX_LENGTH 255`
- 9.11.1.4 `#define NAS_IMSI_M_S1_LENGTH 7`
- 9.11.1.5 `#define NAS_IMSI_M_S2_LENGTH 3`
- 9.11.1.6 `#define NAS_MAX_DESCRIPTION_LENGTH 255`
- 9.11.1.7 `#define NAS_MAX_NUM_NETWORKS 30`
- 9.11.1.8 `#define NAS_MAX_PHY_CA_AGG_CELL_ARRRY_SIZE 255`

9.11.1.9 `#define NAS_MAX_SCC_RX_INFO_INSTANCES 255`

9.11.1.10 `#define NAS_MAX_SLQS_NAS_PCI_INFO_LENGTH 255`

9.11.1.11 `#define NAS_MAX_SLQS_NAS_PCI_INFO_PLMN_LENGTH 255`

9.11.1.12 `#define NAS_MCC_MNC_INST_LENGTH 255`

9.11.1.13 `#define NAS_NAM_NAME_LENGTH 12`

9.11.1.14 `#define NAS_OTA_MESSAGE_MAX_BUF_SIZE 2048`

9.11.1.15 `#define NAS_PLMN_LENGTH 3`

9.11.1.16 `#define NAS_SERVING_SYSTEM_INFO_MAX_RADIO_INTERFACE_LIST 255`

## 9.11.2 Typedef Documentation

9.11.2.1 `typedef nas_BandPrefInfoTlv nas_LTEBandPrefExtTlv`

Contain the LTE Band Preference Extended.

- See [nas\\_BandPrefInfoTlv](#) for more information.

9.11.2.2 `typedef nas_BandPrefInfoTlv nas_NR5gBandPrefTlv`

Contain the NR5G Band Preference.

- See [nas\\_BandPrefInfoTlv](#) for more information.

9.11.2.3 `typedef unpack_result_t unpack_nas_InitiateDomainAttach_t`

9.11.2.4 `typedef unpack_result_t unpack_nas_SetACCOLC_t`

9.11.2.5 `typedef unpack_result_t unpack_nas_SetCDMANetworkParameters_t`

9.11.2.6 `typedef unpack_result_t unpack_nas_SetLURejectCallback_t`

9.11.2.7 `typedef unpack_result_t unpack_nas_SetRFInfoCallback_t`

9.11.2.8 `typedef unpack_result_t unpack_nas_SLQSConfigSigInfo_t`

9.11.2.9 `typedef unpack_result_t unpack_nas_SLQSInitiateNetworkRegistration_t`

9.11.2.10 `typedef unpack_result_t unpack_nas_SLQSNasConfigSigInfo2_t`

9.11.2.11 `typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterExt_t`

9.11.2.12 `typedef unpack_result_t unpack_nas_SLQSNasIndicationRegisterV2_t`

9.11.2.13 `typedef unpack_result_t unpack_nas_SLQSNASSeteDRXParams_t`

9.11.2.14 `typedef unpack_result_t unpack_nas_SLQSNasSwiIndicationRegister_t`



9.11.2.15 `typedef unpack_result_t unpack_nas_SLQSNASSwiSetChannelLock_t`

9.11.2.16 `typedef unpack_result_t unpack_nas_SLQSSetBandPreference_t`

9.11.2.17 `typedef unpack_result_t unpack_nas_SLQSSetSignalStrengthsCallback_t`

9.11.2.18 `typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPref_t`

9.11.2.19 `typedef unpack_result_t unpack_nas_SLQSSetSysSelectionPrefExt_t`

9.11.2.20 `typedef unpack_result_t unpack_nas_SLQSSwiPSDetach_t`

### 9.11.3 Enumeration Type Documentation

9.11.3.1 `enum LITE_TYPE_OF_SERVICE_DOMAIN`

Enumerator

***LITE\_SYS\_SRV\_DOMAIN\_NO\_SRV***  
***LITE\_SYS\_SRV\_DOMAIN\_CS\_ONLY***  
***LITE\_SYS\_SRV\_DOMAIN\_PS\_ONLY***  
***LITE\_SYS\_SRV\_DOMAIN\_CS\_PS***  
***LITE\_SYS\_SRV\_DOMAIN\_CAMPED***

9.11.3.2 `enum LITEQMI_LTE_BAND_CLASS_E_UTRA_OPERATING_BAND`

Enumerator

***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_1***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_2***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_3***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_4***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_5***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_6***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_7***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_8***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_9***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_10***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_11***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_12***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_13***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_14***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_17***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_33***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_34***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_35***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_36***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_37***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_38***  
***eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_39***

*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_40*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_18*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_19*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_20*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_21*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_24*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_25*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_41*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_42*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_43*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_23*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_26*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_32*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_125*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_126*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_127*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_28*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_29*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_30*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_66*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_250*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_46*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_27*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_31*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_71*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_47*  
*eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_48*

#### 9.11.3.3 enum LITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB

Enumerator

*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6*  
*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15*  
*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25*  
*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50*  
*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75*  
*eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100*

#### 9.11.3.4 enum LITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE

Enumerator

*eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED*  
*eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED*  
*eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED*

## 9.11.3.5 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***

## 9.11.3.6 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***

## 9.11.4 Function Documentation

## 9.11.4.1 int pack\_nas\_GetACCOLC ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Retrieves information about the access overload class (ACCOLC) 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

## 9.11.4.2 int pack\_nas\_GetANAAAAuthenticationStatus ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

AN-AAA authentication status of the device 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

#### 9.11.4.3 int pack\_nas\_GetCDMANetworkParameters ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Get CDMA Network Parameters 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

#### 9.11.4.4 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>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

#### 9.11.4.5 int pack\_nas\_GetHomeNetwork3GPP2 ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Get Home Network 3GPP2 Parameters 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

#### 9.11.4.6 int pack\_nas\_GetNetworkPreference ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Gets the network registration preference pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

#### 9.11.4.7 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>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

#### 9.11.4.8 int pack\_nas\_GetServingNetwork ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Pack provides information about the system that provides service to the device.

## 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

#### 9.11.4.9 int pack\_nas\_GetServingNetworkCapabilities ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Pack returns information regarding the data capabilities of the system that currently provides service to the device.

**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

#### 9.11.4.10 int pack\_nas\_GetSignalStrengths ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReq*, uint16\_t \* *pLen* )

get signal strengths pack. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15-C\_05\_xx\_xx and all EM74xx firmware versions. Please use [pack\\_nas\\_SLQSNasGetSigInfo\(\)](#) for new firmware versions and new modules

**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

#### 9.11.4.11 int pack\_nas\_InitiateDomainAttach ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_InitiateDomainAttach\_t \* *pReqParam* )

Initiates a domain attach/detach of the device 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>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

9.11.4.12 `int pack_nas_PerformNetworkScan ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack performs scan for available networks and scans for RAT info as well.

#### 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

9.11.4.13 `int pack_nas_PerformNetworkScanPCI ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_PerformNetworkScanPCI_t * pReqParam )`

Pack performs scan for available networks and scans for LTE RAT info (specific Network or Scan Type).

#### 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>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

#### Note

Device Supported: MC75xx, EM75xx and WP76xx.

9.11.4.14 `int pack_nas_SetACCOLC ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SetACCOLC_t reqParam )`

Pack Sets the access overload class (ACCOLC)

#### 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 Parameters

**Returns**

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

**See Also**

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

**9.11.4.15** int pack\_nas\_SetCDMANetworkParameters ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_nas\_SetCDMANetworkParameters\_t \* pReqParam )

CDMA Network Parameters pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReqParam	request Parameters

**Returns**

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

**See Also**

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

**9.11.4.16** int pack\_nas\_SetLURejectCallback ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, uint8\_t \* pBenable )

Pack set registration reject Callback

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pBenable	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

**9.11.4.17** int pack\_nas\_SetNetworkPreference ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_nas\_SetNetworkPreference\_t \* reqArg )

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 [pack\\_nas\\_SLQSSetSysSelection-Pref\(\)](#) 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

9.11.4.18 `int pack_nas_SetRFInfoCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pBenable )`

Pack Set RF Band Information callback.

## 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>pBenable</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

9.11.4.19 `int pack_nas_SLQSSConfigSigInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSConfigSigInfo_t * pReqParam )`

Config Sig Info Parameters 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>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

9.11.4.20 `int pack_nas_SLQSGetErrorRate ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get Error Rate Parameters 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

9.11.4.21 `int pack_nas_SLQSGetHomeNetwork ( 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>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

9.11.4.22 `int pack_nas_SlqsGetLTECphyCAInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack get carrier aggregation event information.

**Parameters**

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length

9.11.4.23 `int pack_nas_SLQSGetNetworkTime ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack Get Network Time.

**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

9.11.4.24 `int pack_nas_SLQSGetOperatorNameData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Operator Name Data Parameters 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

9.11.4.25 `int pack_nas_SLQSGetPLMNName ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSGetPLMNName_t * reqArg )`

Pack get operator name for specified network.

#### 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

9.11.4.26 `int pack_nas_SLQSGetServingSystem ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack queries information regarding the system that currently provides service.

#### 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

9.11.4.27 `int pack_nas_SLQSGetServingSystemV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack queries information regarding the system that currently provides service.

#### 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

9.11.4.28 `int pack_nas_SLQSGetSignalStrength ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint16_t reqMask )`

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 [pack\\_nas\\_SLQSNasGetSigInfo\(\)](#) 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>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

9.11.4.29 `int pack_nas_SLQSGetSysInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack get system information.

#### 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

9.11.4.30 `int pack_nas_SLQSGetSysInfoV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack get system information.

## 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

9.11.4.31 `int pack_nas_SLQSGetSysSelectionPref ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Queries the different system selection preferences of the device 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

9.11.4.32 `int pack_nas_SLQSGetSysSelectionPrefExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Queries the different system selection preferences of the device 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

9.11.4.33 `int pack_nas_SLQSGetSysSelectionPrefExtV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Queries the different system selection preferences of the device 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

9.11.4.34 `int pack_nas_SLQSInitiateNetworkRegistration ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSInitiateNetworkRegistration_t * pReqParam )`

Pack initiates a network registration. This API is deprecated on MC73xx/EM73xx modules since firmware version S-WI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use [pack\\_nas\\_SLQSSetSysSelectionPrefExt\(\)](#) 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>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

9.11.4.35 `int pack_nas_SLQSNasConfigSigInfo2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasConfigSigInfo2_t * pReqParam )`

Pack sets the signal strength reporting thresholds

## 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>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

9.11.4.36 `int pack_nas_SLQSNasGet3GPP2Subscription ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasGet3GPP2Subscription_t * reqArg )`

3GPP2 Subscription Parameters 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 prarmeters

**Returns**

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

**See Also**

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

9.11.4.37 `int pack_nas_SLQSNasGetCellLocationInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack retrieves cell location-related information.

**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

9.11.4.38 `int pack_nas_SLQSNasGetCellLocationInfoV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack retrieves cell location-related information.

**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

9.11.4.39 `int pack_nas_SLQSNASGeteDRXParams ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get eDRX Parameters 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

9.11.4.40 `int pack_nas_SLQSNASGeteDRXParamsExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNASGeteDRXParamsExt_t * pReqParam )`

Get eDRX Parameters 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>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

9.11.4.41 `int pack_nas_SLQSNASGetForbiddenNetworks ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get forbidden networks 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

9.11.4.42 int pack\_nas\_SLQSNasGetHDRColorCode ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

HDR Color Code Parameters 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

9.11.4.43 int pack\_nas\_SLQSNasGetRFInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets radio band/channel information 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

9.11.4.44 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>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

9.11.4.45 int pack\_nas\_SLQSNasGetTxRxInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_nas\_SLQSNasGetTxRxInfo\_t \* *reqArg* )

Get Tx Rx Info Parameters 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 prarmeters

**Returns**

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

**See Also**

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

9.11.4.46 int pack\_nas\_SLQSNasIndicationRegisterExt ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_nas\_SLQSNasIndicationRegisterExt\_t \* *pReqParam* )

Pack used to Registers/De-registers for different NAS (Network access service) indications.

**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>pReqParam</i>	request Parameters

**Returns**

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

**Note**

This function 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.

**See Also**

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

9.11.4.47 `int pack_nas_SLQSNasIndicationRegisterV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasIndicationRegisterV2_t * pReqParam )`

Pack used to Registers/De-registers for different NAS (Network access service) indications.

#### 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>pReqParam</i>	request Parameters

#### Returns

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

#### Note

This function 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.

#### See Also

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

9.11.4.48 `int pack_nas_SLQSNASSeteDRXParams ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNASSeteDRXParams_t * pReqParam )`

Set eDRX Parameters 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>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

9.11.4.49 `int pack_nas_SLQSNASSwiGetChannelLock ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Get Channel Lock Parameters 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

**9.11.4.50** `int pack_nas_SLQSNasSwiIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNasSwiIndicationRegister_t * pReqParam )`

Pack sets the registration state for different QMI\_NAS SWI indications.

**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>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

**9.11.4.51** `int pack_nas_SLQSNasSwiModemStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

This function requests the device to return the current status of modem.

**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

**9.11.4.52** `int pack_nas_SLQSNASSwiSetChannelLock ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSNASSwiSetChannelLock_t * pReqParam )`

Set Channel Lock Parameters 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>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

9.11.4.53 `int pack_nas_SLQSPerformNetworkScanV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack performs scan for available networks and scans for RAT info as well.

## 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

9.11.4.54 `int pack_nas_SLQSSetBandPreference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetBandPreference_t * pReqParam )`

Pack sets the different system selection preferences of the device.

## 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>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

9.11.4.55 `int pack_nas_SLQSSetSignalStrengthsCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSignalStrengthsCallback_t * pReqParam )`

Pack set signal strength thresholds callback.

#### 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>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

9.11.4.56 `int pack_nas_SLQSSetSysSelectionPref ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPref_t * pReqParam )`

Sets the different system selection preferences of the device 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>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

9.11.4.57 `int pack_nas_SLQSSetSysSelectionPrefExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_nas_SLQSSetSysSelectionPrefExt_t * pReqParam )`

Sets the different system selection preferences of the device 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>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

9.11.4.58 `int pack_nas_SLQSSwiGetHDRPersonality ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

HDR Personality Parameters 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

9.11.4.59 `int pack_nas_SLQSSwiGetHDRProtSubtype ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

HDR Protocol Sub type Parameters 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

9.11.4.60 `int pack_nas_SLQSSwiGetHRPDStats ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

HRPD Stats Parameters 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

9.11.4.61 `int pack_nas_SLQSSwiGetLteCQI ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack fetch CQI parameters for LTE 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

**Returns**

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

**See Also**

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

9.11.4.62 `int pack_nas_SLQSSwiGetLteSccRxInfo ( pack_qmi_t * pCtx, uint8_t * pReq, uint16_t * pLen )`

get LTE Secondary carrier Rx signal level information 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

9.11.4.63 `int pack_nas_SLQSSwiNetworkDebug ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Network Debug Parameters 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

9.11.4.64 int pack\_nas\_SLQSSwiPSDetach ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_nas\_SLQSSwiPSDetach\_t \* *pReqParam* )

PS Connection Detach Parameters 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>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

9.11.4.65 int unpack\_nas\_GetACCOLC ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_GetACCOLC\_t \* *pOutput* )

Retrieves information about the access overload class (ACCOLC) 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

9.11.4.66 int unpack\_nas\_GetANAAAAuthenticationStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_GetANAAAAuthenticationStatus\_t \* *pOutput* )

AN-AAA authentication status of the device 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

9.11.4.67 `int unpack_nas_GetCDMANetworkParameters ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetCDMANetworkParameters_t * pOutput )`

Get CDMA Network Parameters unpack

**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

9.11.4.68 `int unpack_nas_GetHomeNetwork ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetHomeNetwork_t * pOutput )`

Retrieves information about the home network of the device 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

9.11.4.69 `int unpack_nas_GetHomeNetwork3GPP2 ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetHomeNetwork3GPP2_t * pOutput )`

Get Home Network 3GPP2 Parameters 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

**9.11.4.70** `int unpack_nas_GetNetworkPreference ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetNetworkPreference_t * pOutput )`

Gets the network registration preference unpack

**Parameters**

in	<i>pResp</i>	qmi resp from modem
in	<i>respLen</i>	qmi resp 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

**9.11.4.71** `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

**9.11.4.72** `int unpack_nas_GetServingNetwork ( uint8_t * pResp, uint16_t respLen, unpack_nas_GetServingNetwork_t * pOutput )`

Unpack provides information about the system that provides service to 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 [unpack\\_nas\\_SLQSGetSysInfo\(\)](#) for new firmware versions and new modules

**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

**9.11.4.73** int unpack\_nas\_GetServingNetworkCapabilities ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_GetServingNetworkCapabilities\_t \* *pOutput* )

Unpack returns information regarding the data capabilities of the system that currently provides service to the device.

**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

**9.11.4.74** int unpack\_nas\_GetSignalStrengths ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_GetSignalStrengths\_t  
\* *pOutput* )

Returns the available signal strengths ( in dBm ) as measured by the device in an array. This will also provides the corresponding radio interfaces in an array. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use [unpack\\_nas\\_SLQSNasGetSig-Info\(\)](#) for new firmware versions and new modules

**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

9.11.4.75 `int unpack_nas_InitiateDomainAttach ( uint8_t * pResp, uint16_t respLen, unpack_nas_InitiateDomainAttach_t * pOutput )`

Initiates a domain attach/detach of the device 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

9.11.4.76 `int unpack_nas_PerformNetworkScan ( uint8_t * pResp, uint16_t respLen, unpack_nas_PerformNetworkScan_t * pOutput )`

Unpack performs scan for available networks and scans for RAT info as well.

#### 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

9.11.4.77 `int unpack_nas_SetACCOLC ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetACCOLC_t * pOutput )`

Unpack Sets the access overload class (ACCOLC)

#### 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

9.11.4.78 `int unpack_nas_SetCDMANetworkParameters ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SetCDMANetworkParameters_t * pOutput )`

CDMA Network Parameters 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

9.11.4.79 `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

9.11.4.80 `int unpack_nas_SetEventReportInd ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetEventReportInd_t *  
pOutput )`

Unpack set event report indication.

#### 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

9.11.4.81 `int unpack_nas_SetLURjectCallback ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetLURjectCallback_t * pOutput )`

Unpack set registration reject Callback

## 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

9.11.4.82 `int unpack_nas_SetNasLTECphyCalndCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetNasLTECphyCalndCallback_ind_t * pOutput )`

Unpack set LTE PHY CA indication callback.

## 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

9.11.4.83 `int unpack_nas_SetNetworkPreference ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetNetworkPreference_t * pOutput )`

Sets the network registration preference unpack

## Parameters

in	<i>pResp</i>	qmi resp from modem
in	<i>respLen</i>	qmi resp 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

9.11.4.84 `int unpack_nas_SetRFInfoCallback ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetRFInfoCallback_t * pOutput )`

Unack Set RF Band Information callback.

**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

9.11.4.85 `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

9.11.4.86 `int unpack_nas_SetServingSystemCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SetServingSystemCallback_ind_t * pOutput )`

Set Serving system 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

9.11.4.87 int unpack\_nas\_SLQSCfigSigInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSCfigSigInfo\_t \* *pOutput* )

Config Sig Info Parameters 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

9.11.4.88 int unpack\_nas\_SLQSGetErrorRate ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetErrorRate\_t \* *pOutput* )

Get Error Rate Parameters 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

9.11.4.89 int unpack\_nas\_SLQSGetHomeNetwork ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetHomeNetwork\_t \* *pOutput* )

Retrieves information about the home network of the device 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

**9.11.4.90** int unpack\_nas\_SlqsGetLTCphyCAInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SlqsGetLTCphyCAInfo\_t \* *pOutput* )

Unack get carrier aggregation event information.

**Parameters**

in	<i>pResp</i>	qmi resp from modem
in	<i>respLen</i>	qmi resp 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

**9.11.4.91** int unpack\_nas\_SLQSGetNetworkTime ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetNetworkTime\_t \* *pOutput* )

Unpack get network time.

**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

**9.11.4.92** int unpack\_nas\_SLQSGetOperatorNameData ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetOperatorNameData\_t \* *pOutput* )

Operator Name Data Parameters 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

9.11.4.93 int unpack\_nas\_SLQSGetPLMNName ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetPLMNName\_t \* *pOutput* )

unpack get operator name for specified network.

## 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

9.11.4.94 int unpack\_nas\_SLQSGetServingSystem ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetServingSystem\_t \* *pOutput* )

Unack queries information regarding the system that currently provides service.

## 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

9.11.4.95 int unpack\_nas\_SLQSGetServingSystemV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSGetServingSystemV2\_t \* *pOutput* )

Unack queries information regarding the system that currently provides service.

## 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

9.11.4.96 `int unpack_nas_SLQSGetSignalStrength ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSignalStrength_t * pOutput )`

Unpack get signal strength. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C-05\_xx\_xx and all EM74xx firmware versions. Please use [unpack\\_nas\\_SLQSNasGetSigInfo\(\)](#) for new firmware versions and new modules

**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

9.11.4.97 `int unpack_nas_SLQSGetSysInfo ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfo_t * pOutput )`

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The function [unpack\\_nas\\_SLQSGetServingSystem\\_t\(\)](#) reports similar NAS information, but it is deprecated. Please refer to the header description of [unpack\\_nas\\_SLQSGetServingSystem\\_t\(\)](#) for more information.

**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

**Note**

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.

**See Also**

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

9.11.4.98 `int unpack_nas_SLQSGetSysInfoV2 ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysInfoV2_t * pOutput )`

Provides the system information. This API is preferred when trying to get the service status info and serving system info. The function [unpack\\_nas\\_SLQSGetServingSystem\\_t\(\)](#) reports similar NAS information, but it is deprecated. Please refer to the header description of [unpack\\_nas\\_SLQSGetServingSystem\\_t\(\)](#) for more information.

#### 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

#### Note

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.

#### See Also

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

9.11.4.99 `int unpack_nas_SLQSGetSysSelectionPref ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysSelectionPref_t * pOutput )`

Queries the different system selection preferences of the device 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

9.11.4.100 `int unpack_nas_SLQSGetSysSelectionPrefExt ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSGetSysSelectionPrefExt_t * pOutput )`

Queries the different system selection preferences of the device 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

**9.11.4.101** int unpack\_nas\_SLQSGetSysSelectionPrefExtV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t \* *pOutput* )

Queries the different system selection preferences of the device 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

**9.11.4.102** int unpack\_nas\_SLQSIInitiateNetworkRegistration ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSIInitiateNetworkRegistration\_t \* *pOutput* )

Unpack initiates a network registration. This API is deprecated on MC73xx/EM73xx modules since firmware version SWI9X15C\_05\_xx\_xx\_xx and all EM74xx firmware versions. Please use [unpack\\_nas\\_SLQSSetSysSelectionPrefExt\(\)](#) instead for new firmware versions and new modules.

**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

**9.11.4.103** int unpack\_nas\_SLQSNasConfigSigInfo2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasConfigSigInfo2\_t \* *pOutput* )

Unpack sets the signal strength reporting thresholds

## 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

9.11.4.104 int unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t \* *pOutput* )

unpack nas eDRX change Info indication

## 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

9.11.4.105 int unpack\_nas\_SLQSNasGet3GPP2Subscription ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasGet3GPP2Subscription\_t \* *pOutput* )

3GPP2 Subscription Parameters unpack.

## 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

9.11.4.106 `int unpack_nas_SLQSNasGetCellLocationInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasGetCellLocationInfo_t * pOutput )`

Unpack retrieves cell location-related information.

#### 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

9.11.4.107 `int unpack_nas_SLQSNasGetCellLocationInfoV2 ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasGetCellLocationInfoV2_t * pOutput )`

Unpack retrieves cell location-related information.

#### 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

9.11.4.108 `int unpack_nas_SLQSNASGeteDRXParams ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNASGeteDRXParams_t * pOutput )`

Get eDRX Parameters 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

9.11.4.109 `int unpack_nas_SLQSNASGeteDRXParamsExt ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNASGeteDRXParamsExt_t * pOutput )`

Get eDRX Parameters 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

9.11.4.110 `int unpack_nas_SLQSNASGetForbiddenNetworks ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNASGetForbiddenNetworks_t * pOutput )`

Get forbidden networks 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

9.11.4.111 `int unpack_nas_SLQSNasGetHDRColorCode ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasGetHDRColorCode_t * pOutput )`

HDR Color Code Parameters 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

9.11.4.112 int unpack\_nas\_SLQSNasGetRFInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSNasGetRFInfo\_t \* *pOutput* )

Gets radio band/channel information 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

9.11.4.113 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

**Note**

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

**See Also**

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

9.11.4.114 int unpack\_nas\_SLQSNasGetTxRxInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSNasGetTxRxInfo\_t \* *pOutput* )

Get Tx Rx Info Parameters unpack

## 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

9.11.4.115 int unpack\_nas\_SLQSNasIndicationRegisterExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasIndicationRegisterExt\_t \* *pOutput* )

Unpack used to Registers/De-registers for different NAS (Network access service) indications.

## 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

9.11.4.116 int unpack\_nas\_SLQSNasIndicationRegisterV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasIndicationRegisterV2\_t \* *pOutput* )

Unpack used to Registers/De-registers for different NAS (Network access service) indications.

## 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

9.11.4.117 `int unpack_nas_SLQSNasNetworkRejectCallback_Ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasNetworkRejectCallback_Ind_t * pOutput )`

unpack nas Network Reject Callback indication

#### 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

9.11.4.118 `int unpack_nas_SLQSNasNetworkTimeCallBack_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasNetworkTimeCallBack_ind_t * pOutput )`

Unpack Network time callback indication.

#### 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

9.11.4.119 `int unpack_nas_SLQSNasRFBandInfoCallback_Ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasRFBandInfoCallback_Ind_t * pOutput )`

unpack nas RF Band Info Callback indication

#### 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

9.11.4.120 `int unpack_nas_SLQSNASSeteDRXParams ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSNASSeteDRXParams_t * pOutput )`

Set eDRX Parameters 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

9.11.4.121 `int unpack_nas_SLQSNasSigInfoCallback_ind ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSNasSigInfoCallback_ind_t * pOutput )`

Unpack Signal Information callback indication.

## 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

9.11.4.122 `int unpack_nas_SLQSNASSwiGetChannelLock ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSNASSwiGetChannelLock_t * pOutput )`

Get Channel Lock Parameters 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

**9.11.4.123** int unpack\_nas\_SLQSNasSwiIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasSwiIndicationRegister\_t \* *pOutput* )

unpack sets the registration state for different QMI\_NAS SWI indications.

**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

**9.11.4.124** int unpack\_nas\_SLQSNasSwiModemStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSNasSwiModemStatus\_t \* *pOutput* )

This function requests the device to return the current status of modem.

**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

**9.11.4.125** 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

9.11.4.126 `int unpack_nas_SLQSNASSwiSetChannelLock ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNASSwiSetChannelLock_t * pOutput )`

Set Channel Lock Parameters 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

9.11.4.127 `int unpack_nas_SLQSNasSysInfoCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSysInfoCallback_ind_t * pOutput )`

Unpack system information callback indication.

## 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

9.11.4.128 `int unpack_nas_SLQSNasTimerCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSNasTimerCallback_ind_t * pOutput )`

Unpack timer callback indication.

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	network timer indication unpacked

**Returns**

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

**See Also**

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

**9.11.4.129** int unpack\_nas\_SLQSPerformNetworkScanV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSPerformNetworkScanV2\_t \* *pOutput* )

Unpack performs scan for available networks and scans for RAT info as well.

**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

**9.11.4.130** int unpack\_nas\_SLQSSetBandPreference ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSSetBandPreference\_t \* *pOutput* )

Unpack sets the different system selection preferences of the device.

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked

**9.11.4.131** int unpack\_nas\_SLQSSetSignalStrengthsCallback ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_nas\_SLQSSetSignalStrengthsCallback\_t \* *pOutput* )

Unpack set signal strength thresholds callback.

**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

**9.11.4.132** `int unpack_nas_SLQSSetSysSelectionPref ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSetSysSelectionPref_t * pOutput )`

Sets the different system selection preferences of the device 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

**9.11.4.133** `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

**9.11.4.134** `int unpack_nas_SLQSSetSysSelectionPrefExt ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSetSysSelectionPrefExt_t * pOutput )`

Sets the different system selection preferences of the device 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

9.11.4.135 `int unpack_nas_SLQSSwiGetHDRPersonality ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRPersonality_t * pOutput )`

HDR Personality Parameters 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

9.11.4.136 `int unpack_nas_SLQSSwiGetHDRProtSubtype ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHDRProtSubtype_t * pOutput )`

HDR Protocol Sub type Parameters 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

9.11.4.137 `int unpack_nas_SLQSSwiGetHRPDStats ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiGetHRPDStats_t * pOutput )`

HRPD Stats Parameters 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

9.11.4.138 int unpack\_nas\_SLQSSwiGetLteCQI ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSSwiGetLteCQI\_t \* *pOutput* )

Unack fetch CQI parameters for LTE data session.

## 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

9.11.4.139 int unpack\_nas\_SLQSSwiGetLteSccRxInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t \* *pOutput* )

get LTE Secondary carrier Rx signal level information 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

9.11.4.140 int unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind\_t \* *pOutput* )

unpack nas SWI Hdr Personality indication

## 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

9.11.4.141 `int unpack_nas_SLQSSwiNetworkDebug ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSwiNetworkDebug_t * pOutput )`

Network Debug Parameters 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

9.11.4.142 `int unpack_nas_SLQSSwiPSDetach ( uint8_t * pResp, uint16_t respLen, unpack_nas_SLQSSwiPSDetach_t  
* pOutput )`

PS Connection Detach Parameters 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

9.11.4.143 `int unpack_nas_SLQSSwiRandIndicatorCallback_Ind ( uint8_t * pResp, uint16_t respLen,  
unpack_nas_SLQSSwiRandIndicatorCallback_Ind_t * pOutput )`

unpack nas SWI Rank indicator indication

**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

9.11.4.144 int unpack\_valid\_nas\_GetCDMANetworkParameters ( uint8\_t \* *pResp*, uint8\_t *u8Info* )

valid CDMA Network Parameters

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>u8Info</i>	value to check <a href="#">unpack_nas_GetCDMANetworkParameters_t</a> param valid. <ul style="list-style-type: none"><li>• 0 - SCI</li><li>• 1 - SCM</li><li>• 2 - RegHomeSID, RegForeignSID and RegForeignNID</li><li>• 3 - ForceRev0</li><li>• 4 - CustomSCP, Protocol ,Broadcast and Application</li><li>• 5 - Roaming</li></ul>

**Returns**

eQCWWAN\_ERR\_NONE on signal info valid, eQCWWAN\_xxx error value otherwise

**See Also**

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

9.11.4.145 int unpack\_valid\_nas\_SLQSGetServingSystem ( uint8\_t \* *pResp*, uint8\_t *u8Info* )

valid queries information regarding the system that currently provides service

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>u8Info</i>	value to check <a href="#">unpack_nas_SLQSGetServingSystem_t</a> param valid. <ul style="list-style-type: none"> <li>• 0 - RoamIndicatorVal</li> <li>• 1 - DataSrvCapabilities</li> <li>• 2 - CurrentPLMN</li> <li>• 3 - SystemID and NetworkID</li> <li>• 4 - BaseStationID, BaseStationLatitude and BasestationLongitude</li> <li>• 5 - RoamingIndicatorList</li> <li>• 6 - DefaultRoamInd</li> <li>• 7 - Gpp2TimeZone</li> <li>• 8 - CDMA_P_Rev</li> <li>• 9 - GppTimeZone</li> <li>• 10 - GppNetworkDSTAdjustment</li> <li>• 11 - Lac</li> <li>• 12 - CellID</li> <li>• 13 - ConcSvcInfo</li> <li>• 14 - PRLInd</li> <li>• 15 - DTMIInd</li> <li>• 16 - DetailedSvcInfo</li> <li>• 17 - CDMASystemInfoExt</li> <li>• 18 - HdrPersonality</li> <li>• 19 - TrackAreaCode</li> <li>• 20 - CallBarStatus</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on signal info valid, eQCWWAN\_xxx error value otherwise

## See Also

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

9.11.4.146 int unpack\_valid\_nas\_SLQSGetSignalStrength ( uint8\_t \* *pResp*, uint8\_t *u8Info* )

valid queries information regarding the system that currently provides service

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>u8Info</i>	value to check <a href="#">unpack_nas_SLQSGetSignalStrength_t</a> param valid. <ul style="list-style-type: none"> <li>• 0 - rxSignalStrengthList and rxSignalStrengthListLen</li> <li>• 1 - ecioList and ecioListLen</li> <li>• 2 - lo</li> <li>• 3 - sinr</li> <li>• 4 - errorRateListLen, errorRateList</li> <li>• 5 - rsrqInfo</li> <li>• 6 - ltesnr</li> <li>• 7 - ltersrp</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on signal info valid, eQCWWAN\_xxx error value otherwise

## See Also

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

9.11.4.147 int unpack\_valid\_nas\_SLQSNasGetSigInfo ( uint8\_t \* *pResp*, uint8\_t *u8Info* )

valid get sig info

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>u8Info</i>	value to check <a href="#">unpack_nas_SLQSNasGetSigInfo_t</a> param valid. <ul style="list-style-type: none"> <li>• 0 - CDMASInfo</li> <li>• 1 - HDRSSInfo.</li> <li>• 2 - GSMSSInfo.</li> <li>• 3 - WCDMAInfo.</li> <li>• 4 - LTESSInfo.</li> </ul>

## Returns

eQCWWAN\_ERR\_NONE on signal info valid, eQCWWAN\_xxx error value otherwise

## See Also

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

## 9.12 pds.h File Reference

### Data Structures

- struct [unpack\\_pds\\_GetPDSSState\\_t](#)
- struct [pack\\_pds\\_SetPDSSState\\_t](#)
- struct [unpack\\_pds\\_GetPDSDDefaults\\_t](#)
- struct [pack\\_pds\\_SetPDSDDefaults\\_t](#)
- struct [unpack\\_pds\\_GetPortAutomaticTracking\\_t](#)
- struct [pack\\_pds\\_SetPortAutomaticTracking\\_t](#)
- struct [pack\\_pds\\_StartPDSTrackingSessionExt\\_t](#)
- struct [pack\\_pds\\_PDSInjectTimeReference\\_t](#)
- struct [unpack\\_pds\\_GetXTRAAutomaticDownload\\_t](#)
- struct [pack\\_pds\\_SetXTRAAutomaticDownload\\_t](#)
- struct [unpack\\_pds\\_GetXTRANetwork\\_t](#)
- struct [pack\\_pds\\_SetXTRANetwork\\_t](#)
- struct [unpack\\_pds\\_GetXTRAValidity\\_t](#)
- struct [unpack\\_pds\\_GetServiceAutomaticTracking\\_t](#)
- struct [pack\\_pds\\_SetServiceAutomaticTracking\\_t](#)
- struct [pack\\_pds\\_ResetPDSDData\\_t](#)
- struct [pack\\_pds\\_SLQSSetAGPSConfig\\_t](#)
- struct [pack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference\\_t](#)

- struct [pack\\_pds\\_SLQSGetAGPSConfig\\_t](#)
- struct [unpack\\_pds\\_SLQSGetAGPSConfig\\_t](#)
- struct [pack\\_pds\\_SLQSPDSInjectPositionData\\_t](#)
- struct [unpack\\_pds\\_SLQSGetGPSSStateInfo\\_t](#)
- struct [pack\\_pds\\_SLQSSetPositionMethodState\\_t](#)
- struct [pack\\_pds\\_SetEventReportCallback\\_t](#)
- struct [unpack\\_pds\\_SetEventReport\\_Ind\\_t](#)
- struct [unpack\\_pds\\_SetPdsState\\_Ind\\_t](#)

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPDSSState\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPDSDDefaults\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPortAutomaticTracking\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_StartPDSTrackingSessionExt\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_StopPDSTrackingSession\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_PDSInjectTimeReference\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetXTRAAutomaticDownload\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetXTRANetwork\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_ForceXTRADownload\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetServiceAutomaticTracking\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_ResetPDSDData\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSSetAGPSConfig\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSInjectPositionData\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSDeterminePosition\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSSetPositionMethodState\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetEventReportCallback\\_t](#)

## Functions

- int [pack\\_pds\\_GetPDSSState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_pds\\_GetPDSSState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_GetPDSSState\\_t](#) \*pOutput)
- int [pack\\_pds\\_SetPDSSState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_pds\\_SetPDSSState\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_SetPDSSState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetPDSSState\\_t](#) \*pOutput)
- int [pack\\_pds\\_GetPDSDDefaults](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_pds\\_GetPDSDDefaults](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_GetPDSDDefaults\\_t](#) \*pOutput)
- int [pack\\_pds\\_SetPDSDDefaults](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_pds\\_SetPDSDDefaults\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_SetPDSDDefaults](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetPDSDDefaults\\_t](#) \*pOutput)
- int [pack\\_pds\\_GetPortAutomaticTracking](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_pds\\_GetPortAutomaticTracking](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_GetPortAutomaticTracking\\_t](#) \*pOutput)
- int [pack\\_pds\\_SetPortAutomaticTracking](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_pds\\_SetPortAutomaticTracking\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_SetPortAutomaticTracking](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetPortAutomaticTracking\\_t](#) \*pOutput)
- int [pack\\_pds\\_StartPDSTrackingSessionExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_pds\\_StartPDSTrackingSessionExt\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_StartPDSTrackingSessionExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_StartPDSTrackingSessionExt\\_t](#) \*pOutput)



- [int pack\\_pds\\_StopPDSTrackingSession](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_StopPDSTrackingSession](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_StopPDS-TrackingSession\\_t](#) \*pOutput)
- [int pack\\_pds\\_PDSInjectTimeReference](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds\\_P-DSInjectTimeReference\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_PDSInjectTimeReference](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_PDSInjectTime-Reference\\_t](#) \*pOutput)
- [int pack\\_pds\\_GetXTRAAutomaticDownload](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_GetXTRAAutomaticDownload](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_GetXTRA-AutomaticDownload\\_t](#) \*pOutput)
- [int pack\\_pds\\_SetXTRAAutomaticDownload](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_-pds\\_SetXTRAAutomaticDownload\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SetXTRAAutomaticDownload](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SetXTRA-AutomaticDownload\\_t](#) \*pOutput)
- [int pack\\_pds\\_GetXTRANetwork](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_GetXTRANetwork](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_GetXTRANetwork\\_t](#) \*p-Output)
- [int pack\\_pds\\_SetXTRANetwork](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds\\_SetXTRA-Network\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SetXTRANetwork](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SetXTRANetwork\\_t](#) \*p-Output)
- [int pack\\_pds\\_GetXTRAValidity](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_GetXTRAValidity](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_GetXTRAValidity\\_t](#) \*p-Output)
- [int pack\\_pds\\_ForceXTRADownload](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_ForceXTRADownload](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_ForceXTRA-Download\\_t](#) \*pOutput)
- [int pack\\_pds\\_GetServiceAutomaticTracking](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_GetServiceAutomaticTracking](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_GetService-AutomaticTracking\\_t](#) \*pOutput)
- [int pack\\_pds\\_SetServiceAutomaticTracking](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_-pds\\_SetServiceAutomaticTracking\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SetServiceAutomaticTracking](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SetService-AutomaticTracking\\_t](#) \*pOutput)
- [int pack\\_pds\\_ResetPDSData](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds\\_ResetPDS-Data\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_ResetPDSData](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_ResetPDSData\\_t](#) \*pOutput)
- [int pack\\_pds\\_SLQSSetAGPSConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds\\_SLQ-SSetAGPSConfig\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SLQSSetAGPSConfig](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SLQSSetAGPS-Config\\_t](#) \*pOutput)
- [int pack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*p-  
Len, [pack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_S-LQSPDSInjectAbsoluteTimeReference\\_t](#) \*pOutput)
- [int pack\\_pds\\_SLQSGetAGPSConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds\\_SLQ-SGetAGPSConfig\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SLQSGetAGPSConfig](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SLQSGetAGPS-Config\\_t](#) \*pOutput)
- [int pack\\_pds\\_SLQSPDSInjectPositionData](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_pds-\\_SLQSPDSInjectPositionData\\_t](#) \*pReqParam)
- [int unpack\\_pds\\_SLQSPDSInjectPositionData](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SLQSPDS-InjectPositionData\\_t](#) \*pOutput)
- [int pack\\_pds\\_SLQSPDSDeterminePosition](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen)
- [int unpack\\_pds\\_SLQSPDSDeterminePosition](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_pds\\_SLQSPDS-DeterminePosition\\_t](#) \*pOutput)

- int [pack\\_pds\\_SLQSGetGPSStateInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_pds\\_SLQSGetGPSStateInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SLQSGetGPSStateInfo\\_t](#) \*pOutput)
- int [pack\\_pds\\_SLQSSetPositionMethodState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_pds\\_SLQSSetPositionMethodState\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_SLQSSetPositionMethodState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SLQSSetPositionMethodState\\_t](#) \*pOutput)
- int [pack\\_pds\\_SetEventReportCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReq, uint16\_t \*pLen, [pack\\_pds\\_SetEventReportCallback\\_t](#) \*pReqParam)
- int [unpack\\_pds\\_SetEventReportCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetEventReportCallback\\_t](#) \*pOutput)
- int [unpack\\_pds\\_SetEventReport\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetEventReport\\_Ind\\_t](#) \*pOutput)
- int [unpack\\_pds\\_SetPdsState\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_pds\\_SetPdsState\\_Ind\\_t](#) \*pOutput)

### 9.12.1 Typedef Documentation

9.12.1.1 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_ForceXTRADownload\\_t](#)

9.12.1.2 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_PDSInjectTimeReference\\_t](#)

9.12.1.3 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_ResetPDSDData\\_t](#)

9.12.1.4 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetEventReportCallback\\_t](#)

9.12.1.5 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPDSDDefaults\\_t](#)

9.12.1.6 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPDSSState\\_t](#)

9.12.1.7 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetPortAutomaticTracking\\_t](#)

9.12.1.8 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetServiceAutomaticTracking\\_t](#)

9.12.1.9 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetXTRAAutomaticDownload\\_t](#)

9.12.1.10 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SetXTRANetwork\\_t](#)

9.12.1.11 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSDeterminePosition\\_t](#)

9.12.1.12 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSInjectAbsoluteTimeReference\\_t](#)

9.12.1.13 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSPDSInjectPositionData\\_t](#)

9.12.1.14 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSSetAGPSConfig\\_t](#)

9.12.1.15 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_SLQSSetPositionMethodState\\_t](#)

9.12.1.16 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_StartPDSTrackingSessionExt\\_t](#)

9.12.1.17 typedef [unpack\\_result\\_t](#) [unpack\\_pds\\_StopPDSTrackingSession\\_t](#)

### 9.12.2 Function Documentation

### 9.12.2.1 int pack\_pds\_ForceXTRADownload ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Forces the XTRA database to be downloaded to the device 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

### 9.12.2.2 int pack\_pds\_GetPDSDefaults ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Get the default tracking session configuration 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

### 9.12.2.3 int pack\_pds\_GetPDSState ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Get current PDS 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

#### 9.12.2.4 int pack\_pds\_GetPortAutomaticTracking ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Returns the automatic tracking configuration for the NMEA COM port 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

#### 9.12.2.5 int pack\_pds\_GetServiceAutomaticTracking ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

automatic tracking state for the service 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

#### 9.12.2.6 int pack\_pds\_GetXTRAAutomaticDownload ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

XTRA automatic database download configuration 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

### 9.12.2.7 int pack\_pds\_GetXTRANetwork ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

XTRA WWAN network preference parameter 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

### 9.12.2.8 int pack\_pds\_GetXTRValidity ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

XTRA database validity period parameter 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

### 9.12.2.9 int pack\_pds\_PDSInjectTimeReference ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_pds\_PDSInjectTimeReference\_t \* pReqParam )

Inject system time parameters 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>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

9.12.2.10 `int pack_pds_ResetPDSData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_ResetPDSData_t * pReqParam )`

Resets the specified PDS data 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>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

9.12.2.11 `int pack_pds_SetEventReportCallback ( pack_qmi_t * pCtx, uint8_t * pReq, uint16_t * pLen, pack_pds_SetEventReportCallback_t * pReqParam )`

Enables/disables the PDS Event report callback function 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>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

9.12.2.12 `int pack_pds_SetPDSDefaults ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SetPDSDefaults_t * pReqParam )`

Sets the default tracking session configuration 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>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

**9.12.2.13** int pack\_pds\_SetPDSState ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_pds\_SetPDSState\_t \* *pReqParam* )

Sets the PDS 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
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

**9.12.2.14** int pack\_pds\_SetPortAutomaticTracking ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_pds\_SetPortAutomaticTracking\_t \* *pReqParam* )

Sets the automatic tracking configuration for the NMEA COM port 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>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

**9.12.2.15** int pack\_pds\_SetServiceAutomaticTracking ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_pds\_SetServiceAutomaticTracking\_t \* *pReqParam* )

Sets the automatic tracking state for the service 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>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

**9.12.2.16** int pack\_pds\_SetXTRAAutomaticDownload ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_pds\_SetXTRAAutomaticDownload\_t \* *pReqParam* )

Sets the XTRA automatic database download configuration 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>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

**9.12.2.17** int pack\_pds\_SetXTRANetwork ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_pds\_SetXTRANetwork\_t \* *pReqParam* )

Sets the XTRA WWAN network preference 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>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



9.12.2.18 `int pack_pds_SLQSGetAGPSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSGetAGPSConfig_t * pReqParam )`

Gets the PDS AGPS (MS-based) configuration 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>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

9.12.2.19 `int pack_pds_SLQSGetGPSStateInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Queries the MSM GPS server for receiver state information 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

9.12.2.20 `int pack_pds_SLQSPSDeterminePosition ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions 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

9.12.2.21 `int pack_pds_SLQSPDSInjectAbsoluteTimeReference ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSPDSInjectAbsoluteTimeReference_t * pReqParam )`

Injects a absolute time reference into the PDS engine 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>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

9.12.2.22 `int pack_pds_SLQSPDSInjectPositionData ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSPDSInjectPositionData_t * pReqParam )`

Injects position data into the PDS engine 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>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

9.12.2.23 `int pack_pds_SLQSSetAGPSConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSSetAGPSConfig_t * pReqParam )`

Sets the PDS AGPS (MS-based) configuration 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>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

**9.12.2.24** `int pack_pds_SLQSSetPositionMethodState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_SLQSSetPositionMethodState_t * pReqParam )`

Sets the state of positioning methods for the device 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>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

**9.12.2.25** `int pack_pds_StartPDSTrackingSessionExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_pds_StartPDSTrackingSessionExt_t * pReqParam )`

Starts a PDS tracking 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>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

**9.12.2.26** `int pack_pds_StopPDSTrackingSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Stops a PDS tracking 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

**Returns**

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

**See Also**

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

**9.12.2.27** int unpack\_pds\_ForceXTRADownload ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_ForceXTRA-Download\_t \* *pOutput* )

Forces the XTRA database to be downloaded to the device 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

**9.12.2.28** int unpack\_pds\_GetPDSDefaults ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_GetPDSDefaults\_t \* *pOutput* )

Get the default tracking session configuration 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

**9.12.2.29** int unpack\_pds\_GetPDSState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_GetPDSState\_t \* *pOutput* )

Get current PDS 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

9.12.2.30 int unpack\_pds\_GetPortAutomaticTracking ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_GetPortAutomaticTracking\_t \* *pOutput* )

Returns the automatic tracking configuration for the NMEA COM port 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

9.12.2.31 int unpack\_pds\_GetServiceAutomaticTracking ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_GetServiceAutomaticTracking\_t \* *pOutput* )

automatic tracking state for the service 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

9.12.2.32 `int unpack_pds_GetXTRAAutomaticDownload ( uint8_t * pResp, uint16_t respLen,  
unpack_pds_GetXTRAAutomaticDownload_t * pOutput )`

XTRA automatic database download configuration 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

9.12.2.33 `int unpack_pds_GetXTRANetwork ( uint8_t * pResp, uint16_t respLen, unpack_pds_GetXTRANetwork_t *  
pOutput )`

XTRA WWAN network preference parameter 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

9.12.2.34 `int unpack_pds_GetXTRAVality ( uint8_t * pResp, uint16_t respLen, unpack_pds_GetXTRAVality_t *  
pOutput )`

XTRA database validity period parameter 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

9.12.2.35 `int unpack_pds_PDSInjectTimeReference ( uint8_t * pResp, uint16_t respLen,  
unpack_pds_PDSInjectTimeReference_t * pOutput )`

Inject system time parameters 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

9.12.2.36 `int unpack_pds_ResetPDSData ( uint8_t * pResp, uint16_t respLen, unpack_pds_ResetPDSData_t * pOutput )`

Resets the specified PDS data 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

9.12.2.37 `int unpack_pds_SetEventReport_Ind ( uint8_t * pResp, uint16_t respLen, unpack_pds_SetEventReport_Ind_t  
* pOutput )`

PDS 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

**9.12.2.38** int unpack\_pds\_SetEventReportCallback ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_SetEventReportCallback\_t \* *pOutput* )

Enables/disables the PDS Event report callback function 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

**9.12.2.39** int unpack\_pds\_SetPDSDDefaults ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetPDSDDefaults\_t \*  
*pOutput* )

Sets the default tracking session configuration 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

**9.12.2.40** int unpack\_pds\_SetPDSSState ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetPDSSState\_t \* *pOutput* )

Sets the PDS 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

9.12.2.41 int unpack\_pds\_SetPdsState\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetPdsState\_Ind\_t \* *pOutput* )

PDS 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

9.12.2.42 int unpack\_pds\_SetPortAutomaticTracking ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetPortAutomaticTracking\_t \* *pOutput* )

Sets the automatic tracking configuration for the NMEA COM port 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

9.12.2.43 int unpack\_pds\_SetServiceAutomaticTracking ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetServiceAutomaticTracking\_t \* *pOutput* )

Sets the automatic tracking state for the service 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

**9.12.2.44** int unpack\_pds\_SetXTRAAutomaticDownload ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_SetXTRAAutomaticDownload\_t \* *pOutput* )

Sets the XTRA automatic database download configuration 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

**9.12.2.45** int unpack\_pds\_SetXTRANetwork ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SetXTRANetwork\_t \*  
*pOutput* )

Sets the XTRA WWAN network 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

**9.12.2.46** int unpack\_pds\_SLQSGetAGPSConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SLQSGetAGPS-  
Config\_t \* *pOutput* )

Gets the PDS AGPS (MS-based) configuration 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

9.12.2.47 int unpack\_pds\_SLQSGetGPSStateInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SLQSGetGPSStateInfo\_t \* *pOutput* )

Queries the MSM GPS server for receiver state information 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

9.12.2.48 int unpack\_pds\_SLQSPDSDeterminePosition ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SLQSPDSDeterminePosition\_t \* *pOutput* )

Requests the MSM GPS service to obtain the current position for manually controlled tracking sessions 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

9.12.2.49 int unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t \* *pOutput* )

Injects a absolute time reference into the PDS engine 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

**9.12.2.50** int unpack\_pds\_SLQSPDSInjectPositionData ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_SLQSPDSInjectPositionData\_t \* *pOutput* )

Injects position data into the PDS engine 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

**9.12.2.51** int unpack\_pds\_SLQSSetAGPSConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_pds\_SLQSSetAGPS-  
Config\_t \* *pOutput* )

Sets the PDS AGPS (MS-based) configuration 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

**9.12.2.52** int unpack\_pds\_SLQSSetPositionMethodState ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_SLQSSetPositionMethodState\_t \* *pOutput* )

Sets the state of positioning methods for the device 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

**9.12.2.53** int unpack\_pds\_StartPDSTrackingSessionExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_StartPDSTrackingSessionExt\_t \* *pOutput* )

Starts a PDS tracking 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

**9.12.2.54** int unpack\_pds\_StopPDSTrackingSession ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_pds\_StopPDSTrackingSession\_t \* *pOutput* )

Stops a PDS tracking 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

## 9.13 qaGobiApiTableBandClasses.h File Reference

Network Access Service API Band Classes table.

### 9.13.1 Detailed Description

Network Access Service API Band Classes table.

### 9.13.2 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

#### 9.13.2.1 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
- 152 - E-UTRA Operating Band 23
- 153 - E-UTRA Operating Band 26
- 154 - E-UTRA Operating Band 32
- 155 - E-UTRA Operating Band 125
- 156 - E-UTRA Operating Band 126
- 157 - E-UTRA Operating Band 127
- 158 - E-UTRA Operating Band 28
- 159 - E-UTRA Operating Band 29
- 160 - E-UTRA Operating Band 30
- 161 - E-UTRA Operating Band 66
- 162 - E-UTRA Operating Band 250
- 163 - E-UTRA Operating Band 46
- 164 - E-UTRA Operating Band 27
- 165 - E-UTRA Operating Band 31
- 166 - E-UTRA Operating Band 71
- 167 - E-UTRA Operating Band 47
- 168 - E-UTRA Operating Band 48
- 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
- 250 - NR5G Band 1
- 251 - NR5G Band 2

Copyright: © 2018 Sierra Wireless, Inc. all rights reserved

## 9.14 qaGobiApiTableCallControlReturnReasons.h File Reference

Call Control Return Reasons table.

### 9.14.1 Detailed Description

Call Control Return Reasons table.

#### 9.14.2 S1

- 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\_OUTGOINGINTEXTHOME - 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

## 9.15 qaGobiApiTableCallEndReasons.h File Reference

Wireless Data Service Call End Reasons.

### 9.15.1 Detailed Description

Wireless Data Service Call End Reasons.

### 9.15.2 Call end reason codes (Code - Reason)

#### 9.15.2.1 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

#### 9.15.2.2 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

#### 9.15.2.3 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 SNDCP(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

#### 9.15.2.4 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

#### 9.15.2.5 call end reason type

- 1 - Mobile IP
- 2 - Internal
- 3 - Call Manager defined
- 6 - 3GPP specification defined
- 7 - PPP
- 8 - EHRPD
- 9 - IPv6

#### 9.15.2.6 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

#### 9.15.2.7 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
- 227 - INTERNAL\_PDN\_INACTIVITY\_TIMER\_EXPIRED, this error code is returned when PDN inactivity timer expired due to no data transmission in a configurable duration of time
- 228 - MAX\_V4\_CONNECTIONS, this error code is returned when IPv4 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv4 data connections
- 229 - MAX\_V6\_CONNECTIONS, this error code is returned when IPv6 data call bring up is rejected because the UE already maintains the allotted maximum number of IPv6 data connections
- 230 - APN\_MISMATCH, this error code is returned when New PDN bring up is rejected during interface selection because the UE has already allotted the available interfaces for other PDNs
- 231 - IP\_VERSION\_MISMATCH, this error code is returned when New call bring up is rejected because the existing data call IP type does not match the requested IP type
- 232 - DUN\_CALL\_DISALLOWED, this error code is returned when DUN call bring up is rejected because the UE is in eHRPD RAT
- 233 - INVALID\_PROFILE, this error code is returned when Call bring up was requested with an invalid profile
- 234 - INTERNAL\_EPC\_NONEPC\_TRANSITION, this error code is returned when Data call is rejected or brought down because the UE is in transition between EPC and non-EPC RAT
- 235 - INVALID\_PROFILE\_ID, this error code is returned when Call bring up was requested with an invalid profile ID
- 236 - INTERNAL\_CALL\_ALREADY\_PRESENT, this error code is returned when a data call with the same policy is already connected or in the process of bring up on another RmNet instance
- 237 - IFACE\_IN\_USE, this error code is returned when the current IFACE is in use
- 238 - IP\_PDP\_MISMATCH, this error code is returned when a PPP call is attempted on a PDP profile
- 239 - APN\_DISALLOWED\_ON\_ROAMING, this error code is returned when PDN connection to the APN is disallowed on the roaming network. The DS generates DS\_SYS\_EVENT\_3GPP\_ROAMING\_DISALLOWED\_INFO to inform clients of the APNs that must deregister and disconnect (if the PDN is up) while the UE is on the roaming network. Clients must deregister or disconnect if the APN name corresponding to the client is listed in this system event. Clients can register again if the APN name corresponding to the client is not listed in this system event.
- 240 - APN\_PARAM\_CHANGE, this error code is returned when Failure to reestablish the PDN with the changed parameters; when APN-related parameters are changed, the PDN associated with the parameters must be brought down and reestablished with the changed parameters
- 241 - IFACE\_IN\_USE\_CFG\_MATCH, this error code is returned when the IFACE is already in use with a matching configuration
- 242 - NULL\_APN\_DISALLOWED, this error code is returned when a PDN is attempted to be brought up with a NULL APN and when a NULL APN is not supported
- 243 - THERMAL\_MITIGATION, this error code is returned when the thermal level increases and causes calls to be torn down when normal mode of operation is not allowed
- 244 - SUBS\_ID\_MISMATCH, this error code is returned when a new call bring up is rejected due to a mismatch between the subs\_id in the profile and the subs\_id in the ACL policy information
- 245 - DATA\_SETTINGS\_DISABLED, this error code is returned when the PDN connection to a given APN is disallowed because data is disabled from the device user interface (UI) settings
- 246 - DATA\_ROAMING\_SETTINGS\_DISABLED, this error code is returned when the PDN Connection to a given APN is disallowed because data roaming is disabled from the device UI settings and the UE is roaming

- 247 - APN\_FORMAT\_INVALID, this error code is returned when the APN specified in the policy does not follow the format specified in the 3GPP Specification
- 248 - DDS\_CALL\_ABORT, this error code is returned when a DDS switch occurs; the data call is brought down with the DDS abort reason
- 249 - VALIDATION\_FAILURE, this error code is returned when a data call is brought down due to an internal validation failure

#### 9.15.2.8 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
- 535 - 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 - LRRCLTE Radio Resource Control) CONN EST FAILURE CELL RESEL
- 1055 - LRRCLTE Radio Resource Control) CONN EST FAILURE CONFIG FAILURE
- 1056 - LRRCLTE Radio Resource Control) CONN EST FAILURE TIMER EXPIRED
- 1057 - LRRCLTE Radio Resource Control) CONN EST FAILURE LINK FAILURE
- 1058 - LRRCLTE Radio Resource Control) CONN EST FAILURE NOT CAMPED
- 1059 - LRRCLTE Radio Resource Control) CONN EST FAILURE SI FAILURE
- 1060 - LRRCLTE Radio Resource Control) CONN EST FAILURE CONN REJECT
- 1061 - LRRCLTE Radio Resource Control) CONN REL NORMAL
- 1062 - LRRCLTE Radio Resource Control) CONN REL RLF
- 1063 - LRRCLTE Radio Resource Control) CONN REL CRE FAILURE
- 1064 - LRRCLTE Radio Resource Control) CONN REL OOS DURING CRE
- 1065 - LRRCLTE Radio Resource Control) CONN REL ABORTED
- 1066 - LRRCLTE 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

## 9.15.2.9 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 SNDCP FAILURE, PDP context deactivation initiated by the MS or by the Network
- 26 - INSUFFICIENT RESOURCES, this reason is posted to indicate that the network cannot provide the requested service due to insufficient resources
- 27 - MISSING OR UNKNOWN APN, the APN was required and not specified or APN could not be resolved. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 28 - UNKNOWN PDN TYPE, the reason is posted by the network to indicate that the PDN type was not recognized
- 29 - AUTH FAILED, the reason is posted when authentication fails. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 30 - GGSN REJECT, the reason is posted when the request was rejected by Serving GW or PDN GW. In LTE mode of operation, this is a PDN throttling cause code, meaning the UE may end up throttling further requests to the same APN
- 31 - ACTIVATION REJECT, the reason is posted when the request is rejected by the network due to unspecified reasons
- 32 - OPTION NOT SUPPORTED, the reason is posted when UE requested a service not supported by the PLMN
- 33 - OPTION UNSUBSCRIBED, This cause is sent when the MS requests a service option for which it has no subscription
- 34 - OPTION TEMP OOO, service option temporarily out of order, this reason is posted when the network is temporarily out of resources to service the request
- 35 - PTI ALREADY USED, the reason is posted to indicate that PTI (Procedure Transaction Identifier) used in the request is already active via another UE requested procedure
- 36 - REGULAR DEACTIVATION, this reason is posted by the network to initiate a regular release of bearer resources
- 37 - EPS QOS NOT ACCEPTED, this reason is posted by the network to indicate that the QoS requested by the UE could not be accepted
- 38 - NETWORK FAILURE, this reason is posted when an error occurs in the network
- 39 - UMTS REACTIVATION REQ, this reason is posted by the network to request for bearer reactivation. This code may be posted during network congestion
- 40 - FEATURE NOT SUPPORTED, Unsuccessful MBMS context activation requested by the network
- 41 - TFT SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying TFT operation included in the request
- 42 - TFT SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying TFT operation included in the request
- 43 - UNKNOWN PDP CONTEXT, the reason is posted when the bearer identity (or linked bearer identity) in the request is invalid (or inactive)
- 44 - FILTER SEMANTIC ERROR, the reason is posted by the network to indicate semantic error(s) in specifying packet filter(s) associated with a TFT
- 45 - FILTER SYNTAX ERROR, the reason is posted by the network to indicate syntactic error(s) in specifying packet filter(s) associated with a TFT

- 46 - PDP WITHOUT ACTIVE TFT, the reason is posted by the network when UW requested more than one PDP connection without TFT
- 50 - IPV4 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 50, PDN type IPv4 only allowed.
- 51 - IPV6 ONLY ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 51, PDN type IPv6 only allowed
- 52 - SINGLE ADDRESS BEARER ONLY, 3GPP equivalent ESM(EPS Session Management) cause code value 52, Single address bearers only allowed. The reason is posted when the network supports single address bearers only, meaning dual IP bearers are not supported
- 53 - ESM INFORMATION NOT RECEIVED, 3GPP equivalent ESM(EPS Session Management) cause code value 53, ESM information not received. The reason is posted by the network to indicate that the PDN connection request was rejected because ESM information was not received
- 54 - PND CONNECTION DOES NOT EXIST, 3GPP equivalent ESM(EPS Session Management) cause code value 54, PDN connection does not exist The reason is posted by the network during handover from a non-3G-PP network to indicate that the MME does not have any information regarding the requested PDN connection
- 55 - MULTIPLE CONNECTION TO SAME PDN NOT ALLOWED, 3GPP equivalent ESM(EPS Session Management) cause code value 55, Multiple PDN connections for a given APN not allowed. The reason is posted by the network to indicate that the UE is already connected to the requested APN via another PDN/PDN connection
- 81 - INVALID TRANSACTION ID, the reason is posted by the network to indicate that the PTI used in the request is unassigned or reserved
- 95 - MESSAGE INCORRECT SEMANTIC, the reason is posted by the network to indicate receipt of an invalid message
- 96 - INVALID MANDATORY INFO, the reason is posted by the network to indicate receipt of a message with semantic error in a mandatory information element
- 97 - MESSAGE TYPE UNSUPPORTED, the reason is posted by the network to indicate receipt of a message that is either undefined or defined but not implemented by the equipment sending this ESM cause
- 98 - MSG TYPE NONCOMPATIBLE STATE, the reason is posted by the network to indicate receipt of a message type that cannot be handled in the current network protocol state
- 99 - UNKNOWN INFO ELEMENT, the reason is posted by the network to indicate receipt of a message that includes an information element that is either not defined or defined but not implemented by the equipment sending the ESM cause
- 100 - CONDITIONAL IE ERROR, the reason is posted by the network to indicate receipt of a message that includes a syntactically incorrect information element. This message is ignored by the network.
- 101 - MSG AND PROTOCOL STATE UNCOMPATIBLE, the reason is posted by the network to indicate receipt of a message that cannot be handled in the current network protocol state
- 111 - PROTOCOL ERROR, the reason is posted by the network to indicate a protocol error when no other error applies
- 112 - APN TYPE CONFLICT
- 113 - INVALID PROXY-CALL SESSION CONTROL FUNCTION ADDRESS



#### 9.15.2.10 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

#### 9.15.2.11 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

#### 9.15.2.12 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

## 9.16 qaGobiApiTableCarrierCodes.h File Reference

Carrier Codes table.

### 9.16.1 Detailed Description

Carrier Codes table.

#### 9.16.2 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

## 9.17 qaGobiApiTableCodingScheme.h File Reference

Data Coding Scheme.

### 9.17.1 Detailed Description

Data Coding Scheme.

### 9.17.2 S1

#### 9.17.2.1 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

### 9.17.3 Coding Group Bits 7..4(0001)

#### 9.17.3.1 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.

### 9.17.4 Coding Group Bits 7..4(0010)

#### 9.17.4.1 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

### 9.17.5 Coding Group Bits 7..4(0011)

#### 9.17.5.1 use of bits 3..0

- 0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS

### 9.17.6 Coding Group Bits 7..4(01xx)

#### 9.17.6.1 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

### 9.17.7 Coding Group Bits 7..4(1001)

#### 9.17.7.1 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

### 9.17.8 Coding Group Bits 7..4(1010..1101)

#### 9.17.8.1 Reserved coding groups

### 9.17.9 Coding Group Bits 7..4(1110)

#### 9.17.9.1 Defined by the WAP Forum

### 9.17.10 Coding Group Bits 7..4 (1111)

#### 9.17.10.1 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

## 9.18 qaGobiApiTableGpsCapabilityCodes.h File Reference

Position Determination Service API GPS Capability Codes.

### 9.18.1 Detailed Description

Position Determination Service API GPS Capability Codes.

#### 9.18.2 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

## 9.19 qaGobiApiTablePowerModes.h File Reference

Device Management Service API Power Modes table.

### 9.19.1 Detailed Description

Device Management Service API Power Modes table.

#### 9.19.2 Power Modes (Value - Description)

- 0 - Online (default)
- 1 - Low power (airplane) mode
- 2 - Factory test mode
- 3 - Offline
- 4 - Resetting
- 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

## 9.20 `qaGobiApiTableRadioInterfaces.h` File Reference

Network Access Service API Radio Interfaces table.

### 9.20.1 Detailed Description

Network Access Service API Radio Interfaces table.

### 9.20.2 Radio interface

#### 9.20.2.1 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
- 9 - TD-SCDMA
- 10 - LTE-M1
- 11 - LTE-NB1
- 12 - NR5G

Copyright: © 2011 Sierra Wireless, Inc. all rights reserved

## 9.21 `qaGobiApiTableRegionCodes.h` File Reference

Region Codes table.



### 9.21.1 Detailed Description

Region Codes table.

### 9.21.2 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

## 9.22 qaGobiApiTableServiceOptions.h File Reference

Voice Service Options.

### 9.22.1 Detailed Description

Voice Service Options.

### 9.22.2 Service Option codes (Code - Reason)

#### 9.22.2.1 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

## 9.23 qaGobiApiTableSupServiceInfoClasses.h File Reference

Voice Supplementary Service Information Classes.

### 9.23.1 Detailed Description

Voice Supplementary Service Information Classes.

### 9.23.2 Supplementary Service Information Classes (Value - Service Class)

- 0X00 - CLASS\_NONE
- 0X01 - CLASS\_VOICE
- 0X02 - CLASS\_DATA
- 0X04 - CLASS\_FAX
- 0X08 - CLASS\_SMS
- 0X10 - CLASS\_DATA\_CIRCUITSYNC
- 0X20 - CLASS\_DATA\_CIRCUIT\_ASYNC
- 0X40 - CLASS\_PACKET\_ACCESS
- 0X80 - CLASS\_PAD\_ACCESS

Copyright: © 2012 Sierra Wireless, Inc. all rights reserved

## 9.24 qaGobiApiTableSwiAudio.h File Reference

Swi Audio related tables.

### 9.24.1 Detailed Description

Swi Audio related tables.

### 9.24.2 ACDB Device (Device ID - description)

- 0 - Vehicle HF
- 1 - Handset
- 2 - TTY
- 3 - USB
- 4 - NA

### 9.24.3 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

## 9.25 qaGobiApiTableSwiOMADMSessionStatus.h File Reference

Session Status table.

### 9.25.1 Detailed Description

Session Status table.

### 9.25.2 OMA DM Session Status (Session Status - Meaning - Usage)

- 0x0000 - Successful - Session succeeded
- 0x0001 - Break - Session succeeded
- 0x0002 - Large Object Handled - Session succeeded
- 0x0003 - No status - Session succeeded
- 0x0004 - No more commands - Session succeeded
- 0x0005 - User cancel - Session cancelled
- 0x0100-0x01FF - General errors
- 0x0200-0x02FF - Syncml errors
- 0x0300-0x03FF - Authentication errors
- 0x0400-0x04FF - Protocol errors
- 0x0500-0x05FF - Tree errors (DM Only)
- 0x0600-0x06FF - Not applicable
- 0x0700-0x07FF - Trigger errors
- 0x0800-0x08FF - FUMO errors
- 0x0900-0x09FF - Communication errors
- 0x0A00-0x0AFF - Parsing errors
- 0x0B00-0x0CFF - Not applicable
- 0x7F00-0x7F12 - Insignia errors
- 0x7F13 - Illegal text - Text received contains illegal characters
- 0x7F14 - Download failure - Failed to download FOTA image
- 0x7F15 - Empty session - Session ran successfully, but no information was updated
- 0x7F16 - Factory reset successful - Factory reset succeeded
- 0x7F17 - Factory reset fail - Factory reset failed

Copyright: © 2017 Sierra Wireless, Inc. all rights reserved

## 9.26 qaGobiApiTableSwiOMADMUpdateCompleteStatus.h File Reference

Update Complete Status table.

### 9.26.1 Detailed Description

Update Complete Status table.

### 9.26.2 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 TNDIS ) - 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

## 9.27 qaGobiApiTableVoiceCallEndReasons.h File Reference

Voice Service Call and supplementary services end reasons.

### 9.27.1 Detailed Description

Voice Service Call and supplementary services end reasons.

### 9.27.2 Voice Call and supplementary services end reason codes (Code - Reason)

#### 9.27.2.1 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



### 9.27.2.2 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

### 9.27.2.3 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

#### 9.27.2.4 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

#### 9.27.2.5 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

#### 9.27.2.6 reject causes

- 226 - Timer T303 is expired
- 227 - CNM MM release is pending

#### 9.27.2.7 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

#### 9.27.2.8 reject causes

- 237 - Service option is not supported

#### 9.27.2.9 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

## 9.28 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\_EFFECT,
  - 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 }

```

## 9.28.1 Enumeration Type Documentation

### 9.28.1.1 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 - Support for QMI message is 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\_FW\_INVALID\_SLOT\_INDEX** 0xE80F - invalid slot index

**eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_IMAGE\_NOT\_SIGNED** 0xE810 - image not signed

**eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_INVALID\_CERT\_CHAIN** 0xE811 - invalid certificate chain

**eQCWWAN\_ERR\_SWIIM\_FW\_TOO\_MANY\_FILES** 0xE812 - too many files

**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

## 9.28.1.2 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**

## 9.29 qos.h File Reference

## Data Structures

- struct [unpack\\_qos\\_SLQSQosGetNetworkStatus\\_t](#)
- struct [pack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_t](#)
- struct [unpack\\_qos\\_SLQSQosSwiReadApnExtraParams\\_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\\_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](#)
- struct [qos\\_BindDataPortPeripheralEndPointID\\_t](#)
- struct [qos\\_BindDataPortMuxID\\_t](#)
- struct [qos\\_BindDataPortSIODDataPort\\_t](#)
- struct [pack\\_qos\\_BindDataPort\\_t](#)
- struct [unpack\\_qos\\_BindDataPort\\_t](#)

## Macros

- [#define LITEQMI\\_MAX\\_QOS\\_FLOW\\_PER\\_APN\\_STATS 10](#)
- [#define LITEQMI\\_MAX\\_QOS\\_FILTERS 25](#)
- [#define LITEQMI\\_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, [unpack\\_qos\\_SLQSSetQosEventCallback\\_t](#) \*pOutput)
- int [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_SLQSSetQosEventCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_qos\\_BindDataPort](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_qos\\_BindDataPort\\_t](#) req)
- int [unpack\\_qos\\_BindDataPort](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_qos\\_BindDataPort\\_t](#) \*pOutput)

## 9.29.1 Macro Definition Documentation

9.29.1.1 `#define LITEQMI_MAX_QOS_FILTERS 25`

9.29.1.2 `#define LITEQMI_MAX_QOS_FLOW_PER_APN_STATS 10`

9.29.1.3 `#define LITEQMI_MAX_QOS_FLOWS 8`

## 9.29.2 Function Documentation

9.29.2.1 int [pack\\_qos\\_BindDataPort](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_qos\\_BindDataPort\\_t](#) *req* )

Function to pack command to Binds a control point to a data port. This maps to PkQmiQosBindDataPort

### Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
in	<i>req</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_BindDataPort_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

9.29.2.2 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

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<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

9.29.2.3 `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

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<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

**9.29.2.4** `int pack_qos_SLQSQosSviReadDataStats ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_qos_SLQSQosSviReadDataStats_t reqParam )`

Function to pack QMI command to query APN data statistics This maps to SLQSQosSviReadDataStats

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qos_SLQSQosSviReadDataStats_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

**9.29.2.5** `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

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
-----	-------------	---



in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<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

9.29.2.6 `int unpack_qos_BindDataPort ( uint8_t * pResp, uint16_t respLen, unpack_qos_BindDataPort_t * pOutput )`

Function to unpack the response to Binds a control point to a data port. This maps to UpkQmiQosBindDataPort

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_qos_BindDataPort_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

9.29.2.7 `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

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<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

**9.29.2.8** `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

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<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

**9.29.2.9** `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

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
----	--------------	---

in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_qos_SLQSSQoSReadDataStats_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

**9.29.2.10** int unpack\_qos\_SLQSSetQosEventCallback ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_qos\_SLQSSetQosEventCallback\_t \* *pOutput* )

Function to unpack enable QoS event indications command's response This maps to SLQSSetQosEventCallback

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_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

**9.29.2.11** 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**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>

out	<i>pOutput</i>	<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

**9.29.2.12** 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**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<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

9.29.2.13 `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

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<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

9.29.2.14 `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

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pOutput</i>	<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

## 9.30 rms.h File Reference

### Data Structures

- struct [unpack\\_rms\\_GetSMSWake\\_t](#)
- struct [pack\\_rms\\_SetSMSWake\\_t](#)
- struct [unpack\\_rms\\_SetSMSWake\\_t](#)

### Macros

- #define [\\_\\_LITEQMI\\_RMS\\_H\\_\\_](#)

### Functions

- int [pack\\_rms\\_GetSMSWake](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_rms\\_GetSMSWake](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_rms\\_GetSMSWake\\_t](#) \*pOutput)
- int [pack\\_rms\\_SetSMSWake](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_rms\\_SetSMSWake\\_t](#) \*reqArg)
- int [unpack\\_rms\\_SetSMSWake](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_rms\\_SetSMSWake\\_t](#) \*pOutput)

### 9.30.1 Macro Definition Documentation

#### 9.30.1.1 #define \_\_LITEQMI\_RMS\_H\_\_

### 9.30.2 Function Documentation

#### 9.30.2.1 int pack\_rms\_GetSMSWake ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Get SMS Wake 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

9.30.2.2 `int pack_rms_SetSMSWake ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_rms_SetSMSWake_t * reqArg )`

Set SMS Wake 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

9.30.2.3 `int unpack_rms_GetSMSWake ( uint8_t * pResp, uint16_t respLen, unpack_rms_GetSMSWake_t * pOutput )`

Get SMS Wake 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

9.30.2.4 `int unpack_rms_SetSMSWake ( uint8_t * pResp, uint16_t respLen, unpack_rms_SetSMSWake_t * pOutput )`

Set SMS Wake unpack

## 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

## 9.31 sar.h File Reference

**Data Structures**

- struct [unpack\\_sar\\_SLQSGetRfSarState\\_t](#)
- struct [pack\\_sar\\_SLQSSetRfSarState\\_t](#)

**Typedefs**

- typedef [unpack\\_result\\_t](#) [unpack\\_sar\\_SLQSSetRfSarState\\_t](#)

**Functions**

- int [pack\\_sar\\_SLQSGetRfSarState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sar\\_SLQSGetRfSarState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sar\\_SLQSGetRfSarState\\_t](#) \*pOutput)
- int [pack\\_sar\\_SLQSSetRfSarState](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sar\\_SLQSSetRfSarState\\_t](#) \*reqArg)
- int [unpack\\_sar\\_SLQSSetRfSarState](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sar\\_SLQSSetRfSarState\\_t](#) \*pOutput)

### 9.31.1 Typedef Documentation

#### 9.31.1.1 typedef unpack\_result\_t unpack\_sar\_SLQSSetRfSarState\_t

### 9.31.2 Function Documentation

#### 9.31.2.1 int pack\_sar\_SLQSGetRfSarState ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Gets the specified RF SAR 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



9.31.2.2 `int pack_sar_SLQSSetRfSarState ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sar_SLQSSetRfSarState_t * reqArg )`

Sets the specified RF SAR 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
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

9.31.2.3 `int unpack_sar_SLQSGetRfSarState ( uint8_t * pResp, uint16_t respLen, unpack_sar_SLQSGetRfSarState_t * pOutput )`

Gets the specified RF SAR 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

9.31.2.4 `int unpack_sar_SLQSSetRfSarState ( uint8_t * pResp, uint16_t respLen, unpack_sar_SLQSSetRfSarState_t * pOutput )`

Sets the specified RF SAR 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

## 9.32 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\\_SLQSModifySMSStatus\\_t](#)
- struct [unpack\\_sms\\_SLQSModifySMSStatus\\_t](#)
- struct [pack\\_sms\\_SLQSDeleteSMS\\_t](#)
- struct [unpack\\_sms\\_SLQSDeleteSMS\\_t](#)
- struct [pack\\_sms\\_SendSMS\\_t](#)
- struct [unpack\\_sms\\_SendSMS\\_t](#)
- struct [pack\\_sms\\_SetNewSMSCallback\\_t](#)
- struct [unpack\\_sms\\_SetNewSMSCallback\\_t](#)
- struct [SMSMTMessageInfo](#)
- struct [newMTMessageTlv](#)
- struct [SMSTransferRouteMTMessageInfo](#)
- struct [transferRouteMessageTlv](#)
- struct [SMSMessageModelInfo](#)
- struct [messageModeTlv](#)
- struct [SMSEtwsMessageInfo](#)
- struct [SMSEtwsMessageTlv](#)
- struct [SMSEtwsPlmnInfo](#)
- struct [eTWSPLMNInfoTlv](#)
- struct [SMSCAddressInfo](#)
- struct [SMSCAddressTlv](#)
- struct [SMSONIMSInfo](#)
- struct [SMSONIMSTlv](#)
- struct [unpack\\_sms\\_SetNewSMSCallback\\_ind\\_t](#)
- struct [unpack\\_sms\\_SLQSWmsMemoryFullCallBack\\_ind\\_t](#)
- struct [unpack\\_sms\\_GetSMSCAddress\\_t](#)
- struct [pack\\_sms\\_SetSMSCAddress\\_t](#)
- struct [unpack\\_sms\\_SetSMSCAddress\\_t](#)
- struct [pack\\_sms\\_SaveSMS\\_t](#)
- struct [unpack\\_sms\\_SaveSMS\\_t](#)
- struct [sms\\_BroadcastConfig](#)
- struct [sms\\_qaQmi3GPPBroadcastCfgInfo](#)
- struct [sms\\_CDMABroadcastConfig](#)
- struct [sms\\_qaQmi3GPP2BroadcastCfgInfo](#)
- struct [unpack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t](#)
- struct [pack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t](#)
- struct [pack\\_sms\\_SLQSSetSmsBroadcastConfig\\_t](#)
- struct [unpack\\_sms\\_SLQSSetSmsBroadcastConfig\\_t](#)
- struct [pack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t](#)
- struct [unpack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t](#)
- struct [sms\\_transLayerInfo](#)
- struct [sms\\_getTransLayerInfo](#)

- struct [unpack\\_sms\\_SLQSGetTransLayerInfo\\_t](#)
- struct [sms\\_getTransNWRegInfo](#)
- struct [unpack\\_sms\\_SLQSGetTransNWRegInfo\\_t](#)
- struct [sms\\_getIndicationReg](#)
- struct [unpack\\_sms\\_SLQSGetIndicationRegister\\_t](#)
- struct [sms\\_setIndicationReg](#)
- struct [pack\\_sms\\_SLQSSetIndicationRegister\\_t](#)
- struct [unpack\\_sms\\_SLQSSetIndicationRegister\\_t](#)
- struct [sms\\_routeEntry](#)
- struct [sms\\_setRoutesReq](#)
- struct [pack\\_sms\\_SLQSSmsSetRoutes\\_t](#)
- struct [unpack\\_sms\\_SLQSSmsSetRoutes\\_t](#)
- struct [sms\\_msgProtocolResp](#)
- struct [unpack\\_sms\\_SLQSSmsGetMessageProtocol\\_t](#)
- struct [sms\\_maxStorageSizeReq](#)
- struct [sms\\_maxStorageSizeResp](#)
- struct [pack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t](#)
- struct [unpack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t](#)
- struct [sms\\_messageWaitingInfoContent](#)
- struct [sms\\_getMsgWaitingInfo](#)
- struct [unpack\\_sms\\_SLQSGetMessageWaiting\\_t](#)
- struct [sms\\_sendAsynsmsParams](#)
- struct [pack\\_sms\\_SLQSSendAsyncSMS\\_t](#)
- struct [unpack\\_sms\\_SLQSSendAsyncSMS\\_t](#)
- struct [pack\\_sms\\_SLQSSetSmsStorage\\_t](#)
- struct [unpack\\_sms\\_SLQSSetSmsStorage\\_t](#)
- struct [unpack\\_sms\\_SLQSSwiGetSMSStorage\\_t](#)
- struct [unpack\\_sms\\_SLQSTransLayerInfoCallback\\_ind\\_t](#)
- struct [unpack\\_sms\\_SLQSNWRegInfoCallback\\_ind\\_t](#)
- struct [unpack\\_sms\\_SLQSWmsMessageWaitingCallBack\\_ind\\_t](#)
- struct [unpack\\_sms\\_SLQSWmsAsyncRawSendCallBack\\_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
- [#define SMSC\\_TYPE\\_LEN](#) 0x03
- [#define SMS\\_CONFIG\\_LEN](#) 0x05
- [#define SMS\\_MAX\\_SMS\\_ROUTES](#) 0x0A
- [#define SMS\\_NUM\\_OF\\_SET](#) 0xFF

## Enumerations

- enum [eqmiCbKsetStatus](#) {  
[LITEQMI\\_QMI\\_CBK\\_PARAM\\_RESET](#) = 0,  
[LITEQMI\\_QMI\\_CBK\\_PARAM\\_SET](#) = 1,  
[LITEQMI\\_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)
- int [pack\\_sms\\_GetSMSCAddress](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_GetSMSCAddress](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_GetSMSCAddress\\_t](#) \*pOutput)
- int [pack\\_sms\\_SetSMSCAddress](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SetSMSCAddress\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SetSMSCAddress](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SetSMSCAddress\\_t](#) \*pOutput)
- int [pack\\_sms\\_SaveSMS](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SaveSMS\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SaveSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SaveSMS\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSGetSmsBroadcastConfig](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSGetSmsBroadcastConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetSmsBroadcastConfig\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSetSmsBroadcastConfig](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSetSmsBroadcastConfig\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSetSmsBroadcastConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSetSmsBroadcastConfig\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSetSmsBroadcastActivation](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSetSmsBroadcastActivation](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSetSmsBroadcastActivation\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSGetTransLayerInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQSGetTransLayerInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetTransLayerInfo\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSGetTransNWRegInfo](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQSGetTransNWRegInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetTransNWRegInfo\\_t](#) \*pOutput)

- int [pack\\_sms\\_SLQSGetIndicationRegister](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQSGetIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSGetIndicationRegister\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSetIndicationRegister](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSetIndicationRegister\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSetIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSetIndicationRegister\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSmsSetRoutes](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSmsSetRoutes\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSmsSetRoutes](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSmsSetRoutes\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSmsGetMessageProtocol](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQSSmsGetMessageProtocol](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSmsGetMessageProtocol\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSmsGetMaxStorageSize](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSmsGetMaxStorageSize](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSmsGetMaxStorageSize\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQS GetMessageWaiting](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQS GetMessageWaiting](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQS GetMessageWaiting\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSendAsyncSMS](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSendAsyncSMS\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSendAsyncSMS](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSendAsyncSMS\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSetSmsStorage](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_sms\\_SLQSSetSmsStorage\\_t](#) \*reqParam)
- int [unpack\\_sms\\_SLQSSetSmsStorage](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSetSmsStorage\\_t](#) \*pOutput)
- int [pack\\_sms\\_SLQSSwiGetSMSStorage](#) (pack\_qmi\_t \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_sms\\_SLQSSwiGetSMSStorage](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSSwiGetSMSStorage\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSTransLayerInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSTransLayerInfoCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSNWRegInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSNWRegInfoCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSWmsMessageWaitingCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSWmsMessageWaitingCallBack\\_ind\\_t](#) \*pOutput)
- int [unpack\\_sms\\_SLQSWmsAsyncRawSendCallBack\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_sms\\_SLQSWmsAsyncRawSendCallBack\\_ind\\_t](#) \*pOutput)

### 9.32.1 Macro Definition Documentation

9.32.1.1 `#define MAX_CDMA_ENC_MO_TXT_MSG_SIZE 255`

9.32.1.2 `#define MAX_MS_TRANSFER_ROUTE_MSG 256`

9.32.1.3 `#define MAX_MSC_ADDRESS_SIZE 256`

9.32.1.4 `#define MAX_MSE_TWS_MSG 1254`

9.32.1.5 `#define MAX_SMS_LIST_SIZE 255`

9.32.1.6 `#define MAX_SMS_MESSAGE_SIZE 2048`

9.32.1.7 `#define SMS_CONFIG_LEN 0x05`

9.32.1.8 `#define SMS_MAX_SMS_ROUTES 0x0A`

9.32.1.9 `#define SMS_NUM_OF_SET 0xFF`

9.32.1.10 `#define SMSC_TYPE_LEN 0x03`

## 9.32.2 Enumeration Type Documentation

9.32.2.1 `enum eqmiCbkJSetStatus`

Enumerator

***LITEQMI\_QMI\_CBK\_PARAM\_RESET***

***LITEQMI\_QMI\_CBK\_PARAM\_SET***

***LITEQMI\_QMI\_CBK\_PARAM\_NOCHANGE***

## 9.32.3 Function Documentation

9.32.3.1 `int pack_sms_GetSMSCAddress ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets the SMS center address 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

Returns

`eQCWWAN_ERR_NONE` on success, `eQCWWAN_XXX` error value otherwise

See Also

See [qmerrno.h](#) for `eQCWWAN_XXX` error values

9.32.3.2 `int pack_sms_SaveSMS ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SaveSMS_t * reqParam )`

Saves an SMS message to device memory 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

9.32.3.3 `int pack_sms_SendSMS ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SendSMS_t * reqParam )`

send 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

9.32.3.4 `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

9.32.3.5 `int pack_sms_SetSMSCAddress ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SetSMSCAddress_t * reqParam )`

Set the SMS center address 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

**9.32.3.6** `int pack_sms_SLQSDDeleteSMS ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSDDeleteSMS_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

**9.32.3.7** `int pack_sms_SLQSGetIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets registration state of different WMS indications 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

**9.32.3.8** `int pack_sms_SLQSGetMessageWaiting ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets the message waiting information 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

**9.32.3.9** 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	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
	reqParam	packed request

## Returns

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

## See Also

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

**9.32.3.10** int pack\_sms\_SLQSGetSmsBroadcastConfig ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_sms\_SLQSGetSmsBroadcastConfig\_t \* reqParam )

Provides Information about the SMS BroadcastConfiguration pack.

## Parameters

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
	reqParam	packed request

## Returns

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

## See Also

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

**9.32.3.11** 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

9.32.3.12 int pack\_sms\_SLQSGetTransLayerInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets information about the transport layer 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

9.32.3.13 int pack\_sms\_SLQSGetTransNWRegInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets transport layer network registration 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

9.32.3.14 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

**9.32.3.15** int pack\_sms\_SLQSSendAsyncSMS ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_sms\_SLQSSendAsyncSMS\_t \* *reqParam* )

Sends an SMS message for immediate over-the-air transmission 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

**9.32.3.16** int pack\_sms\_SLQSSetIndicationRegister ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_sms\_SLQSSetIndicationRegister\_t \* *reqParam* )

Sets the registration state of different WMS indications 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

9.32.3.17 `int pack_sms_SLQSSetSmsBroadcastActivation ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSetSmsBroadcastActivation_t * reqParam )`

Enables or disables the reception of broadcast SMS messages 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

9.32.3.18 `int pack_sms_SLQSSetSmsBroadcastConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSetSmsBroadcastConfig_t * reqParam )`

Sets the information about the SMS BroadcastConfiguration 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

9.32.3.19 `int pack_sms_SLQSSetSmsStorage ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSetSmsStorage_t * reqParam )`

Sets the SMS Storage on the device 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

**9.32.3.20** `int pack_sms_SLQSSmsGetMaxStorageSize ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSmsGetMaxStorageSize_t * reqParam )`

Get the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available 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

**9.32.3.21** `int pack_sms_SLQSSmsGetMessageProtocol ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Gets the message protocol currently in use for the WMS client 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

**9.32.3.22** `int pack_sms_SLQSSmsSetRoutes ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_sms_SLQSSmsSetRoutes_t * reqParam )`

Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports 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

**9.32.3.23** int pack\_sms\_SLQSSwiGetSMSStorage ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets the current SMS configuration that is applied to all incoming and outgoing messages 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

**9.32.3.24** int unpack\_sms\_GetSMSCAddress ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_GetSMSCAddress\_t \* *pOutput* )

Gets the SMS center address 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

**9.32.3.25** int unpack\_sms\_SaveSMS ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SaveSMS\_t \* *pOutput* )

Saves an SMS message to device memory 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

9.32.3.26 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

9.32.3.27 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
out	<i>Output</i>	unpacked response

## Returns

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

## See Also

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

9.32.3.28 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

9.32.3.29 int unpack\_sms\_SetSMSCAddress ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SetSMSCAddress\_t \* *pOutput* )

Set the SMS center address 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

9.32.3.30 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
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



9.32.3.31 `int unpack_sms_SLQSGetIndicationRegister ( uint8_t * pResp, uint16_t respLen, unpack_sms_SLQSGetIndicationRegister_t * pOutput )`

Gets registration state of different WMS indications 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

9.32.3.32 `int unpack_sms_SLQSGetMessageWaiting ( uint8_t * pResp, uint16_t respLen, unpack_sms_SLQSGetMessageWaiting_t * pOutput )`

Gets the message waiting information 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

9.32.3.33 `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

9.32.3.34 `int unpack_sms_SLQSGetSmsBroadcastConfig ( uint8_t * pResp, uint16_t respLen,  
unpack_sms_SLQSGetSmsBroadcastConfig_t * pOutput )`

Provides Information about the SMS BroadcastConfiguration 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

9.32.3.35 `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

9.32.3.36 `int unpack_sms_SLQSGetTransLayerInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_sms_SLQSGetTransLayerInfo_t * pOutput )`

Gets information about the transport layer 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

**9.32.3.37** int unpack\_sms\_SLQSGetTransNWRegInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSGetTransNWRegInfo\_t \* *pOutput* )

Gets transport layer network registration 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

**9.32.3.38** int unpack\_sms\_SLQSModifySMSStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSModifySMS-  
Status\_t \* *pOutput* )

modify sms 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

**9.32.3.39** int unpack\_sms\_SLQSNWRegInfoCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSNWRegInfoCallback\_ind\_t \* *pOutput* )

Unpack indication about change in transport layer info

## 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

**9.32.3.40** int unpack\_sms\_SLQSSendAsyncSMS ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSSendAsyncSMS\_t \* *pOutput* )

Sends an SMS message for immediate over-the-air transmission 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

**9.32.3.41** int unpack\_sms\_SLQSSetIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSSetIndicationRegister\_t \* *pOutput* )

Sets the registration state of different WMS indications 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

**9.32.3.42** int unpack\_sms\_SLQSSetSmsBroadcastActivation ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSSetSmsBroadcastActivation\_t \* *pOutput* )

Enables or disables the reception of broadcast SMS messages 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

**9.32.3.43** int unpack\_sms\_SLQSSetSmsBroadcastConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSSetSmsBroadcastConfig\_t \* *pOutput* )

Sets the information about the SMS BroadcastConfiguration 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

**9.32.3.44** int unpack\_sms\_SLQSSetSmsStorage ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSSetSms-  
Storage\_t \* *pOutput* )

Sets the SMS Storage on the device 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

**9.32.3.45** int unpack\_sms\_SLQSSmsGetMaxStorageSize ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSSmsGetMaxStorageSize\_t \* *pOutput* )

Get the maximum number of messages that can be stored in the specified memory storage. Also it provides the number of slots currently available 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

**9.32.3.46** int unpack\_sms\_SLQSSmsGetMessageProtocol ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSSmsGetMessageProtocol\_t \* *pOutput* )

Gets the message protocol currently in use for the WMS client 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

**9.32.3.47** int unpack\_sms\_SLQSSmsSetRoutes ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_sms\_SLQSSmsSet-  
Routes\_t \* *pOutput* )

Sets the action performed on SMS message receipt for specified message routes. It also specifies the action performed on SMS receipt of status reports 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

9.32.3.48 int unpack\_sms\_SLQSSwiGetSMSStorage ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSSwiGetSMSStorage\_t \* *pOutput* )

Gets the current SMS configuration that is applied to all incoming and outgoing messages 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

9.32.3.49 int unpack\_sms\_SLQSTransLayerInfoCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t \* *pOutput* )

Unpack indication about change in transport layer info

#### 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

9.32.3.50 int unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t \* *pOutput* )

Unpack indication for sms async raw send

#### 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

9.32.3.51 `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

9.32.3.52 `int unpack_sms_SLQSWmsMessageWaitingCallBack_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_sms_SLQSWmsMessageWaitingCallBack_ind_t * pOutput )`

Unpack indication for message waiting information

**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

## 9.33 swiaudio.h File Reference

**Data Structures**

- struct [pack\\_swiaudio\\_SLQSGetM2MAudioProfile\\_t](#)
- struct [unpack\\_swiaudio\\_SLQSGetM2MAudioProfile\\_t](#)
- struct [pack\\_swiaudio\\_SLQSSetM2MAudioProfile\\_t](#)
- struct [pack\\_swiaudio\\_SLQSGetM2MAudioVolume\\_t](#)
- struct [unpack\\_swiaudio\\_SLQSGetM2MAudioVolume\\_t](#)
- struct [pack\\_swiaudio\\_SLQSSetM2MAudioVolume\\_t](#)
- struct [swiaudio\\_PCMparams](#)



- struct [pack\\_swiaudio\\_SLQSSetM2MAudioAVCFG\\_t](#)
- struct [pack\\_swiaudio\\_SLQSSetM2MAudioLPBK\\_t](#)
- struct [pack\\_swiaudio\\_SLQSGetM2MSpkrGain\\_t](#)
- struct [unpack\\_swiaudio\\_SLQSGetM2MSpkrGain\\_t](#)
- struct [pack\\_swiaudio\\_SLQSSetM2MSpkrGain\\_t](#)
- struct [pack\\_swiaudio\\_SLQSGetM2MAVMute\\_t](#)
- struct [unpack\\_swiaudio\\_SLQSGetM2MAVMute\\_t](#)
- struct [pack\\_swiaudio\\_SLQSSetM2MAVMute\\_t](#)

## Macros

- `#define SWIAUDIO_MAX_LEN_IFACE_TABLE 255`

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAudioProfile\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAudioVolume\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAudioAVCFG\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAudioLPBK\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAudioNVDef\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MSpkrGain\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swiaudio\\_SLQSSetM2MAVMute\\_t](#)

## Functions

- int [pack\\_swiaudio\\_SLQSGetM2MAudioProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSGetM2MAudioProfile\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSGetM2MAudioProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSGetM2MAudioProfile\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAudioProfile](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MAudioProfile\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MAudioProfile](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAudioProfile\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSGetM2MAudioVolume](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSGetM2MAudioVolume\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSGetM2MAudioVolume](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSGetM2MAudioVolume\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAudioVolume](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MAudioVolume\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MAudioVolume](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAudioVolume\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAudioAVCFG](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MAudioAVCFG\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MAudioAVCFG](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAudioAVCFG\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAudioLPBK](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MAudioLPBK\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MAudioLPBK](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAudioLPBK\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAudioNVDef](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swiaudio\\_SLQSSetM2MAudioNVDef](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAudioNVDef\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSGetM2MSpkrGain](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSGetM2MSpkrGain\\_t](#) \*pReqParam)

- int [unpack\\_swiaudio\\_SLQSGetM2MSpkrGain](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSGetM2MSpkrGain\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MSpkrGain](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MSpkrGain\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MSpkrGain](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MSpkrGain\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSGetM2MAVMute](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSGetM2MAVMute\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSGetM2MAVMute](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSGetM2MAVMute\\_t](#) \*pOutput)
- int [pack\\_swiaudio\\_SLQSSetM2MAVMute](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaudio\\_SLQSSetM2MAVMute\\_t](#) \*pReqParam)
- int [unpack\\_swiaudio\\_SLQSSetM2MAVMute](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaudio\\_SLQSSetM2MAVMute\\_t](#) \*pOutput)

### 9.33.1 Macro Definition Documentation

9.33.1.1 `#define SWIAUDIO_MAX_LEN_IFACE_TABLE 255`

### 9.33.2 Typedef Documentation

9.33.2.1 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioAVCFG_t`

9.33.2.2 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioLPBK_t`

9.33.2.3 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioNVDef_t`

9.33.2.4 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioProfile_t`

9.33.2.5 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAudioVolume_t`

9.33.2.6 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MAVMute_t`

9.33.2.7 `typedef unpack_result_t unpack_swiaudio_SLQSSetM2MSpkrGain_t`

### 9.33.3 Function Documentation

9.33.3.1 int [pack\\_swiaudio\\_SLQSGetM2MAudioProfile](#) ( [pack\\_qmi\\_t](#) \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, [pack\\_swiaudio\\_SLQSGetM2MAudioProfile\\_t](#) \* *pReqParam* )

Gets the profile content 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>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

9.33.3.2 `int pack_swiaudio_SLQSGetM2MAudioVolume ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSGetM2MAudioVolume_t * pReqParam )`

Gets the Volume content 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>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

9.33.3.3 `int pack_swiaudio_SLQSGetM2MAVMute ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSGetM2MAVMute_t * pReqParam )`

Gets the AV Mute content 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>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

9.33.3.4 `int pack_swiaudio_SLQSGetM2MSprGain ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSGetM2MSprGain_t * pReqParam )`

Gets the SPKRGAIN content 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>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

**9.33.3.5** `int pack_swiaudio_SLQSSetM2MAudioAVCFG ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioAVCFG_t * pReqParam )`

Sets the AVCFG content 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>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

**9.33.3.6** `int pack_swiaudio_SLQSSetM2MAudioLPBK ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioLPBK_t * pReqParam )`

Sets the LPBK content 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>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

**9.33.3.7** `int pack_swiaudio_SLQSSetM2MAudioNVDef ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Sets the NVDef content 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

**9.33.3.8** `int pack_swiaudio_SLQSSetM2MAudioProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioProfile_t * pReqParam )`

sets an audio profile content 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>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

**9.33.3.9** `int pack_swiaudio_SLQSSetM2MAudioVolume ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAudioVolume_t * pReqParam )`

Sets the Volume content 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>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

9.33.3.10 `int pack_swiaudio_SLQSSetM2MAVMute ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MAVMute_t * pReqParam )`

Sets the AV Mute content 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>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

9.33.3.11 `int pack_swiaudio_SLQSSetM2MSprGain ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaudio_SLQSSetM2MSprGain_t * pReqParam )`

Sets the SPKRGAIN content 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>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

9.33.3.12 `int unpack_swiaudio_SLQSGetM2MAudioProfile ( uint8_t * pResp, uint16_t respLen, unpack_swiaudio_SLQSGetM2MAudioProfile_t * pOutput )`

Gets the profile content 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

9.33.3.13 `int unpack_swiaudio_SLQSGetM2MAudioVolume ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaudio_SLQSGetM2MAudioVolume_t * pOutput )`

Gets the Volume content 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

9.33.3.14 `int unpack_swiaudio_SLQSGetM2MAVMute ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaudio_SLQSGetM2MAVMute_t * pOutput )`

Gets the AV Mute content 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

9.33.3.15 `int unpack_swiaudio_SLQSGetM2MSpkrGain ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaudio_SLQSGetM2MSpkrGain_t * pOutput )`

Gets the SPKRGAIN content 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

**9.33.3.16** int unpack\_swiaudio\_SLQSSetM2MAudioAVCFG ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t \* *pOutput* )

Sets the AVCFG content 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

**9.33.3.17** int unpack\_swiaudio\_SLQSSetM2MAudioLPBK ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAudioLPBK\_t \* *pOutput* )

Sets the LPBK content 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

**9.33.3.18** int unpack\_swiaudio\_SLQSSetM2MAudioNVDef ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAudioNVDef\_t \* *pOutput* )

Sets the NVDef content 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

**9.33.3.19** int unpack\_swiaudio\_SLQSSetM2MAudioProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAudioProfile\_t \* *pOutput* )

sets an audio profile content 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

**9.33.3.20** int unpack\_swiaudio\_SLQSSetM2MAudioVolume ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAudioVolume\_t \* *pOutput* )

Sets the Volume content 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

**9.33.3.21** int unpack\_swiaudio\_SLQSSetM2MAVMute ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MAVMute\_t \* *pOutput* )

Sets the AV Mute content 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

9.33.3.22 int unpack\_swiaudio\_SLQSSetM2MSpkrGain ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaudio\_SLQSSetM2MSpkrGain\_t \* *pOutput* )

Sets the SPKRGAIN content 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

## 9.34 swiavms.h File Reference

**Data Structures**

- struct [pack\\_swiaavms\\_SLQSAVMSStartSession\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSStartSession\\_t](#)
- struct [pack\\_swiaavms\\_SLQSAVMSStopSession\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSStopSession\\_t](#)
- struct [PackSwiAvmsSetSettingsPeriodInfo](#)
- struct [PackSwiAvmsSetSettingsAPNInfo](#)
- struct [PackSwiAvmsSetSettingsConnectionRetryTimers](#)
- struct [pack\\_swiaavms\\_SLQSAVMSSetSettings\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSSetSettings\\_t](#)
- struct [pack\\_swiaavms\\_SLQSAVMSSetSettings\\_v2\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSSetSettings\\_v2\\_t](#)
- struct [PackSwiAVMSSettingsAPNInfo](#)
- struct [PackSwiAVMSSettingsConnectionRetryTimers](#)
- struct [PackSwiAVMSSettingsPeriodsInfo](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSGetSettings\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSGetSettings\\_v2\\_t](#)
- struct [pack\\_swiaavms\\_SLQSAVMSSendSelection\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAVMSSendSelection\\_t](#)
- struct [unpack\\_swiaavms\\_SLQSAvmsSetEventReport\\_t](#)
- struct [UnpackSwiAvmsEventReportBinaryUpdateSessionInfo](#)
- struct [UnpackSwiAvmsEventReportConfig](#)
- struct [UnpackSwiAvmsEventReportNotification](#)
- struct [UnpackSwiAvmsEventReportConnectionRequest](#)

- struct [UnpackSwiAvmsEventReportWAMSPParamChange](#)
- struct [UnpackSwiAvmsEventReportPackageID](#)
- struct [UnpackSwiAvmsEventReportRegStatus](#)
- struct [UnpackSwiAvmsEventReportDataSessionStatus](#)
- struct [UnpackSwiAvmsEventReportSessionType](#)
- struct [UnpackSwiAvmsEventReportHTTPStatus](#)
- struct [unpack\\_swiavms\\_SLQSAVMSSessionGetInfo\\_t](#)
- struct [unpack\\_swiavms\\_SLQSAVMSEventReportInd\\_t](#)

## Macros

- [#define LITEQMI\\_MAX\\_SWIOMA\\_STR\\_LEN](#) 255
- [#define LITEQMI\\_MAX\\_GET\\_SETTINGS\\_AVMS\\_APN\\_STRING\\_LENGTH](#) 49
- [#define LITEQMI\\_MAX\\_GET\\_SETTINGS\\_AVMS\\_UNAME\\_STRING\\_LENGTH](#) 29
- [#define LITEQMI\\_MAX\\_GET\\_SETTINGS\\_AVMS\\_PWD\\_STRING\\_LENGTH](#) 29
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SET\\_SETTING\\_APN\\_LENGTH](#) 49
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SET\\_SETTING\\_UNAME\\_LENGTH](#) 29
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SET\\_SETTING\\_PWD\\_LENGTH](#) 29
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SET\\_SETTING\\_CONNECTION\\_RETRY\\_TIMMERS](#) 8
- [#define MAX\\_AVMS\\_SETTINGS\\_RETRY\\_TIMER\\_NUMBER](#) 8
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SESSIONGETINFO\\_BINARYUPDATESSESSIONINFO\\_VERSION\\_LENGTH](#) 128
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SESSIONGETINFO\\_BINARYUPDATESSESSIONINFO\\_NAME\\_LENGTH](#) 128
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SESSIONGETINFO\\_BINARYUPDATESSESSIONINFO\\_DESC\\_LENGTH](#) 1024
- [#define MAX\\_PACK\\_SWI\\_AVMS\\_SESSIONGETINFO\\_CONFIG\\_ALERT\\_MSG\\_LENGTH](#) 200

## Typedefs

- typedef [unpack\\_swiavms\\_SLQSAVMSStopSession\\_t](#) [unpack\\_swiavms\\_SLQSAVMSStopSession\\_avc2\\_t](#)

## Functions

- int [pack\\_swiavms\\_SLQSAVMSStartSession](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_swiavms\\_SLQSAVMSSstartSession\\_t](#) reqParam)
- int [unpack\\_swiavms\\_SLQSAVMSSstartSession](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_swiavms\\_SLQSAVMSSstartSession\\_t](#) \*pResponse)
- int [pack\\_swiavms\\_SLQSAVMSStopSession](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_swiavms\\_SLQSAVMSStopSession\\_t](#) reqParam)
- int [unpack\\_swiavms\\_SLQSAVMSStopSession](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_swiavms\\_SLQSAVMSStopSession\\_t](#) \*pResponse)
- int [unpack\\_swiavms\\_SLQSAVMSStopSession\\_avc2](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_swiavms\\_SLQSAVMSStopSession\\_t](#) \*pResponse)
- int [pack\\_swiavms\\_SLQSAVMSSetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_swiavms\\_SLQSAVMSSetSettings\\_t](#) reqParam)
- int [unpack\\_swiavms\\_SLQSAVMSSetSettings](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_swiavms\\_SLQSAVMSSetSettings\\_t](#) \*pResponse)
- int [pack\\_swiavms\\_SLQSAVMSSetSettings\\_v2](#) ([pack\\_qmi\\_t](#) \*pCtx, [uint8\\_t](#) \*pReqBuf, [uint16\\_t](#) \*pLen, [pack\\_swiavms\\_SLQSAVMSSetSettings\\_v2\\_t](#) reqParam)
- int [unpack\\_swiavms\\_SLQSAVMSSetSettings\\_v2](#) ([uint8\\_t](#) \*pResp, [uint16\\_t](#) respLen, [unpack\\_swiavms\\_SLQSAVMSSetSettings\\_v2\\_t](#) \*pResponse)

- int [pack\\_swiaavms\\_SLQSAVMSSetSettingsNoAutoRebootField](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaavms\\_SLQSAVMSSetSettings\\_t](#) reqParam)
- int [pack\\_swiaavms\\_SLQSAVMSGetSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swiaavms\\_SLQSAVMSGetSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAVMSGetSettings\\_t](#) \*pResponse)
- int [pack\\_swiaavms\\_SLQSAVMSGetSettings\\_v2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swiaavms\\_SLQSAVMSGetSettings\\_v2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAVMSGetSettings\\_v2\\_t](#) \*pResponse)
- int [pack\\_swiaavms\\_SLQSAVMSSendSelection](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swiaavms\\_SLQSAVMSSendSelection\\_t](#) reqParam)
- int [unpack\\_swiaavms\\_SLQSAVMSSendSelection](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAVMSSendSelection\\_t](#) \*pResponse)
- int [pack\\_swiaavms\\_SLQSAvmsSetEventReport](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swiaavms\\_SLQSAvmsSetEventReport](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAvmsSetEventReport\\_t](#) \*pResponse)
- int [pack\\_swiaavms\\_SLQSAVMSSessionGetInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swiaavms\\_SLQSAVMSSessionGetInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAVMSSessionGetInfo\\_t](#) \*pResponse)
- int [unpack\\_swiaavms\\_SLQSAVMSEventReportInd](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swiaavms\\_SLQSAVMSEventReportInd\\_t](#) \*pResponse)

### 9.34.1 Macro Definition Documentation

- 9.34.1.1 `#define LITEQMI_MAX_GET_SETTINGS_AVMS_APN_STRING_LENGTH 49`
- 9.34.1.2 `#define LITEQMI_MAX_GET_SETTINGS_AVMS_PWD_STRING_LENGTH 29`
- 9.34.1.3 `#define LITEQMI_MAX_GET_SETTINGS_AVMS_UNAME_STRING_LENGTH 29`
- 9.34.1.4 `#define LITEQMI_MAX_SWIOMA_STR_LEN 255`
- 9.34.1.5 `#define MAX_AVMS_SETTINGS_RETRY_TIMER_NUMBER 8`
- 9.34.1.6 `#define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESSESSIONINFO_DESC_LENGTH 1024`
- 9.34.1.7 `#define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESSESSIONINFO_NAME_LENGTH 128`
- 9.34.1.8 `#define MAX_PACK_SWI_AVMS_SESSIONGETINFO_BINARYUPDATESSESSIONINFO_VERSION_LENGTH 128`
- 9.34.1.9 `#define MAX_PACK_SWI_AVMS_SESSIONGETINFO_CONFIG_ALERT_MSG_LENGTH 200`
- 9.34.1.10 `#define MAX_PACK_SWI_AVMS_SET_SETTING_APN_LENGTH 49`
- 9.34.1.11 `#define MAX_PACK_SWI_AVMS_SET_SETTING_CONNECTION_RETRY_TIMMERS 8`
- 9.34.1.12 `#define MAX_PACK_SWI_AVMS_SET_SETTING_PWD_LENGTH 29`
- 9.34.1.13 `#define MAX_PACK_SWI_AVMS_SET_SETTING_UNAME_LENGTH 29`

### 9.34.2 Typedef Documentation

- 9.34.2.1 `typedef unpack\_swiaavms\_SLQSAVMSStopSession\_t unpack\_swiaavms\_SLQSAVMSStopSession\_avc2\_t`

### 9.34.3 Function Documentation

### 9.34.3.1 int pack\_swiavms\_SLQSAVMSGetSettings ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Function to pack AVMS Get settings command This maps to SLQSAVMSSetSettings

#### Parameters

out	pCtx	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	pReqBuf	<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>
in, out	pLen	<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

### 9.34.3.2 int pack\_swiavms\_SLQSAVMSGetSettings\_v2 ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Function to pack AVMS Get settings command This maps to SLQSAVMSSetSettings\_v2 (For AVC2 service)

#### Parameters

out	pCtx	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	pReqBuf	<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>
in, out	pLen	<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

**9.34.3.3** `int pack_swiaavms_SLQSAVMSSendSelection ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaavms_SLQSAVMSSendSelection_t reqParam )`

Function to pack AVMS send selection command This maps to SLQSAVMSSendSelection

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_swiaavms_SLQSAVMSSendSelection_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

**9.34.3.4** `int pack_swiaavms_SLQSAVMSSessionGetInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack AVMS get session info command This maps to SLQSAVMSSessionGetInfo

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>in, out</i>	<i>pLen</i>	<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

#### 9.34.3.5 int pack\_swiavms\_SLQSAvmsSetEventReport ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Function to pack AVMS set event report command This maps to SLQSAVMSSetEventReport

**Parameters**

<i>out</i>	<i>pCtx</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
<i>in, out</i>	<i>pReqBuf</i>	<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>in, out</i>	<i>pLen</i>	<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

#### 9.34.3.6 int pack\_swiavms\_SLQSAVMSSetSettings ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_swiavms\_SLQSAVMSSetSettings\_t reqParam )

Function to pack AVMS Set settings command This maps to SLQSAVMSSetSettings

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_swiaavms_SLQSAVMSSetSettings_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

9.34.3.7 int pack\_swiaavms\_SLQSAVMSSetSettings\_v2 ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t reqParam )

Function to pack AVMS Set settings command This maps to SLQSAVMSSetSettings\_v2 (For AVC2 service)

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_swiaavms_SLQSAVMSSetSettings_v2_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

**9.34.3.8** `int pack_swiavms_SLQSAVMSSetSettingsNoAutoRebootField ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiavms_SLQSAVMSSetSettings_t reqParam )`

Function to pack AVMS Set settings without Auto Reobot field command This maps to SLQSAVMSSetSettings-IgnoreAutoReboot

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>On input, size of pReqBuf</li> <li>On ouptut, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_swiavms_SLQSAVMSSetSettings_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

**9.34.3.9** `int pack_swiavms_SLQSAVMSStartSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiavms_SLQSAVMSStartSession_t reqParam )`

Function to pack Start AVMS session command This maps to SLQSAVMSStartSession

## Parameters

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>See <a href="#">pack_qmi_t</a> for more information</li> </ul>
-----	-------------	---

in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_swiaavms_SLQSAVMSStartSession_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

**9.34.3.10** `int pack_swiaavms_SLQSAVMSStopSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swiaavms_SLQSAVMSStopSession_t reqParam )`

Function to pack cancel AVMS session command This maps to SLQSAVMSStopSession

**Parameters**

out	<i>pCtx</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_qmi_t</a> for more information</li> </ul>
in, out	<i>pReqBuf</i>	<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>
in, out	<i>pLen</i>	<ul style="list-style-type: none"> <li>• On input, size of pReqBuf</li> <li>• On output, number of bytes actually packed</li> </ul>
in	<i>reqParam</i>	<ul style="list-style-type: none"> <li>• See <a href="#">pack_swiaavms_SLQSAVMSStopSession_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

9.34.3.11 `int unpack_swiavms_SLQSAVMSEventReportInd ( uint8_t * pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSEventReportInd_t * pResponse )`

Function to unpack AVMS event report Indication command This maps to SLQSAVMSSetEventReport

## Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swiavms_SLQSAVMSEventReportInd_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

9.34.3.12 `int unpack_swiavms_SLQSAVMSSetSettings ( uint8_t * pResp, uint16_t respLen, unpack_swiavms_SLQSAVMSSetSettings_t * pResponse )`

Function to unpack AVMS set settings command This maps to SLQSAVMSSetSettings

## Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swiavms_SLQSAVMSSetSettings_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

9.34.3.13 `int unpack_swiaavms_SLQSAVMSGetSettings_v2 ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaavms_SLQSAVMSGetSettings_v2_t * pResponse )`

Function to unpack AVMS set settings command This maps to SLQSAVMSGetSettings\_v2 (For AVC2 service)

#### Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiaavms_SLQSAVMSSetSettings_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

9.34.3.14 `int unpack_swiaavms_SLQSAVMSendSelection ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaavms_SLQSAVMSendSelection_t * pResponse )`

Function to unpack AVMS send selection command This maps to SLQSAVMSendSelection

#### Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiaavms_SLQSAVMSendSelection_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

9.34.3.15 `int unpack_swiaavms_SLQSAVMSsessionGetInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_swiaavms_SLQSAVMSsessionGetInfo_t * pResponse )`

Function to unpack AVMS event report Indication command This maps to SLQSAVMSsetEventReport

## Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiavms_SLQSAVMSSessionGetInfo_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

**9.34.3.16** int unpack\_swiavms\_SLQSAvmsSetEventReport ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiavms\_SLQSAvmsSetEventReport\_t \* *pResponse* )

Function to unpack AVMS set event report command This maps to SLQSAVMSSetEventReport

## Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiavms_SLQSAvmsSetEventReport_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

**9.34.3.17** int unpack\_swiavms\_SLQSAVMSSetSettings ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiavms\_SLQSAVMSSetSettings\_t \* *pResponse* )

Function to unpack AVMS Set settings command This maps to SLQSAVMSSetSettings

## Parameters

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
----	--------------	---

in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swiaavms_SLQSAVMSSetSettings_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

9.34.3.18 int unpack\_swiaavms\_SLQSAVMSSetSettings\_v2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t \* *pResponse* )

Function to unpack AVMS Set settings command This maps to SLQSAVMSSetSettings\_v2 (For AVC2 service)

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>See <a href="#">unpack_swiaavms_SLQSAVMSSetSettings_v2_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

9.34.3.19 int unpack\_swiaavms\_SLQSAVMSStartSession ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swiaavms\_SLQSAVMSStartSession\_t \* *pResponse* )

Function to unpack Start AVMS session response from modem This maps to SLQSAVMSStartSession

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>Length of pResp from modem</li> </ul>

out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiavms_SLQSAVMSStartSession_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

**9.34.3.20** int unpack\_swiavms\_SLQSAVMSStopSession ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_swiavms\_SLQSAVMSStopSession\_t \* *pResponse* )

Function to pack stop AVMS session command This maps to SLQSAVMSStopSession

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiavms_SLQSAVMSStopSession_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

**9.34.3.21** int unpack\_swiavms\_SLQSAVMSStopSession\_avc2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_swiavms\_SLQSAVMSStopSession\_t \* *pResponse* )

Function to pack stop AVMS session command This maps to SLQSAVMSStopSession Note: this targets WP760x as the stop response doesn't contains mandatory session\_type(0x01) TLV

**Parameters**

in	<i>pResp</i>	<ul style="list-style-type: none"> <li>• Response from modem</li> </ul>
in	<i>respLen</i>	<ul style="list-style-type: none"> <li>• Length of pResp from modem</li> </ul>
out	<i>pResponse</i>	<ul style="list-style-type: none"> <li>• See <a href="#">unpack_swiavms_SLQSAVMSStopSession_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

## 9.35 SwiDataTypes.h File Reference

SWI data types.

**Data Structures**

- struct [qmTlvResult](#)

**Macros**

- #define [SWI\\_API](#)
- #define [QMI\\_NO\\_LTE\\_FW\\_SUPPORT](#) 0
- #define [QMI\\_TLV\\_PLACEHOLDER](#) 0x8F
- #define [UNUSEDPARAM](#)(x) (void)x
- #define [MEMSET\\_STATIC\\_OUTPUT\\_STRUCT](#)(OUTPUT\_PARA, STRUCT\_OF\_OUTPUT\_PARA) memset(OUTPUT\_PARA,0,sizeof(STRUCT\_OF\_OUTPUT\_PARA));\

**Typedefs**

- typedef unsigned int [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](#)
- typedef unsigned short [qmuint16](#)
- typedef unsigned long [qmulong](#)

### 9.35.1 Detailed Description

SWI data types.

### 9.35.2 Macro Definition Documentation

9.35.2.1 #define [MEMSET\\_STATIC\\_OUTPUT\\_STRUCT](#)( *OUTPUT\_PARA*, *STRUCT\_OF\_OUTPUT\_PARA*  
 ) memset(OUTPUT\_PARA,0,sizeof(STRUCT\_OF\_OUTPUT\_PARA));\

Macro used to memset unpack output paramter. To prevent output parameter is not initialized.



9.35.2.2 `#define QMI_NO_LTE_FW_SUPPORT 0`

9.35.2.3 `#define QMI_TLV_PLACEHOLDER 0x8F`

9.35.2.4 `#define SWI_API`

9.35.2.5 `#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.

### 9.35.3 Typedef Documentation

9.35.3.1 `typedef int BOOL`

9.35.3.2 `typedef unsigned char BYTE`

9.35.3.3 `typedef char CHAR`

9.35.3.4 `typedef float FLOAT`

9.35.3.5 `typedef signed int INT32`

9.35.3.6 `typedef signed char INT8`

9.35.3.7 `typedef const char* LPCSTR`

9.35.3.8 `typedef unsigned short qmuint16`

9.35.3.9 `typedef unsigned long qmulong`

9.35.3.10 `typedef signed short SHORT`

9.35.3.11 `typedef unsigned int ULONG`

9.35.3.12 `typedef unsigned long long ULONGLONG`

9.35.3.13 `typedef unsigned short USHORT`

9.35.3.14 `typedef unsigned short WORD`

## 9.36 swidms.h File Reference

### Data Structures

- struct [unpack\\_swidms\\_SLQSSwiDmsGetUsbNetNum\\_t](#)
- struct [pack\\_swidms\\_SLQSSwiDmsSetUsbNetNum\\_t](#)
- struct [pack\\_swidms\\_SLQSSwiDmsSetMTU\\_t](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsSetMTU\\_t](#)
- struct [swidms\\_mtuSize3gppTlv](#)
- struct [swidms\\_hrpdMTUSizeTlv](#)
- struct [swidms\\_ehrpdMTUSizeTlv](#)
- struct [swidms\\_usbMTUSizeTlv](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsGetMTU\\_t](#)
- struct [swidms\\_interfaceCfgTlv](#)

- struct [swidms\\_supportedIntBitmaskTlv](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsGetUsbComp\\_t](#)
- struct [pack\\_swidms\\_SLQSSwiDmsSetUsbComp\\_t](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsSetUsbComp\\_t](#)
- struct [swidms\\_SwiDmsGetHWWatchdog](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsGetHWWatchdog\\_t](#)
- struct [pack\\_swidms\\_SLQSSwiDmsSetHWWatchdog\\_t](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsSetHWWatchdog\\_t](#)
- struct [unpack\\_swidms\\_SLQSSwiDmsGetSecureInfo\\_t](#)

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_swidms\\_SLQSSwiDmsSetUsbNetNum\\_t](#)

## Functions

- int [pack\\_swidms\\_SLQSSwiDmsGetUsbNetNum](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swidms\\_SLQSSwiDmsGetUsbNetNum](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsGetUsbNetNum\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsSetUsbNetNum](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swidms\\_SLQSSwiDmsSetUsbNetNum\\_t](#) \*pReqParam)
- int [unpack\\_swidms\\_SLQSSwiDmsSetUsbNetNum](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsSetUsbNetNum\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsSetMTU](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swidms\\_SLQSSwiDmsSetMTU\\_t](#) \*reqArg)
- int [unpack\\_swidms\\_SLQSSwiDmsSetMTU](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsSetMTU\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsGetMTU](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swidms\\_SLQSSwiDmsGetMTU](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsGetMTU\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsGetUsbComp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swidms\\_SLQSSwiDmsGetUsbComp](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsGetUsbComp\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsSetUsbComp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swidms\\_SLQSSwiDmsSetUsbComp\\_t](#) \*reqArg)
- int [unpack\\_swidms\\_SLQSSwiDmsSetUsbComp](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsSetUsbComp\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsGetHWWatchdog](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swidms\\_SLQSSwiDmsGetHWWatchdog](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsGetHWWatchdog\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsSetHWWatchdog](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swidms\\_SLQSSwiDmsSetHWWatchdog\\_t](#) \*reqArg)
- int [unpack\\_swidms\\_SLQSSwiDmsSetHWWatchdog](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsSetHWWatchdog\\_t](#) \*pOutput)
- int [pack\\_swidms\\_SLQSSwiDmsGetSecureInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swidms\\_SLQSSwiDmsGetSecureInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swidms\\_SLQSSwiDmsGetSecureInfo\\_t](#) \*pOutput)

## 9.36.1 Typedef Documentation

9.36.1.1 typedef unpack\_result\_t unpack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t

## 9.36.2 Function Documentation

9.36.2.1 int pack\_swidms\_SLQSSwiDmsGetHWWatchdog ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets the hardware watchdog 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

9.36.2.2 int pack\_swidms\_SLQSSwiDmsGetMTU ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Get the preferred MTU Size of the 3GPP, HRPD, EHRPD and USB descriptor Interfaces 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

9.36.2.3 int pack\_swidms\_SLQSSwiDmsGetSecureInfo ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

pack function to get secure boot config and other capabilities

### 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

#### 9.36.2.4 int pack\_swidms\_SLQSSwiDmsGetUsbComp ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Gets the usb interface composition 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

#### 9.36.2.5 int pack\_swidms\_SLQSSwiDmsGetUsbNetNum ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

pack function to get usb net numbers for QMAP configuration

**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

#### 9.36.2.6 int pack\_swidms\_SLQSSwiDmsSetHWWatchdog ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t \* reqArg )

Sets the usb interface config value 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.

**9.36.2.7** `int pack_swidms_SLQSSwiDmsSetMTU ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swidms_SLQSSwiDmsSetMTU_t * reqArg )`

Sets the preferred MTU size for 3GPP, HRPD, EHRPD Interfaces 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

**9.36.2.8** `int pack_swidms_SLQSSwiDmsSetUsbComp ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swidms_SLQSSwiDmsSetUsbComp_t * reqArg )`

Sets the usb interface config value 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. This setting will be effective after modem reboot

**9.36.2.9** `int pack_swidms_SLQSSwiDmsSetUsbNetNum ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swidms_SLQSSwiDmsSetUsbNetNum_t * pReqParam )`

pack function to set usb net numbers for QMAP configuration

**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>pReqParam</i>	request params

**Returns**

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

**See Also**

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

9.36.2.10 int unpack\_swidms\_SLQSSwiDmsGetHWWatchdog ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swidms\_SLQSSwiDmsGetHWWatchdog\_t \* *pOutput* )

Gets the hardware watchdog settings 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

9.36.2.11 int unpack\_swidms\_SLQSSwiDmsGetMTU ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swidms\_SLQSSwiDmsGetMTU\_t \* *pOutput* )

Get the MTU Size of the 3GPP, HRPD, EHRPD and USB descriptor Interfaces 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

9.36.2.12 `int unpack_swidms_SLQSSwiDmsGetSecureInfo ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsGetSecureInfo_t * pOutput )`

unpack function to get secure boot config

#### 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

9.36.2.13 `int unpack_swidms_SLQSSwiDmsGetUsbComp ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsGetUsbComp_t * pOutput )`

Gets the usb interface composition 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

9.36.2.14 `int unpack_swidms_SLQSSwiDmsGetUsbNetNum ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsGetUsbNetNum_t * pOutput )`

unpack function to get usb net numbers for QMAP configuration

#### 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

9.36.2.15 `int unpack_swidms_SLQSSwiDmsSetHWWatchdog ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsSetHWWatchdog_t * pOutput )`

Sets the hardware watchdog settings value 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. This setting will be effective after modem reboot

9.36.2.16 `int unpack_swidms_SLQSSwiDmsSetMTU ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsSetMTU_t * pOutput )`

Sets preferred MTU size for 3GPP, HRPD, EHRPD Interfaces 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

9.36.2.17 `int unpack_swidms_SLQSSwiDmsSetUsbComp ( uint8_t * pResp, uint16_t respLen,  
unpack_swidms_SLQSSwiDmsSetUsbComp_t * pOutput )`

Sets the usb interface config value 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. This setting will be effective after modem reboot

9.36.2.18 int unpack\_swidms\_SLQSSwiDmsSetUsbNetNum ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t \* *pOutput* )

unpack function to set usb net numbers for QMAP configuration

**Parameters**

in	<i>pResp</i>	qmi response
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

## 9.37 swiloc.h File Reference

**Data Structures**

- struct [unpack\\_swiloc\\_SwiLocGetAutoStart\\_t](#)
- struct [pack\\_swiloc\\_SwiLocSetAutoStart\\_t](#)

**Typedefs**

- typedef [unpack\\_result\\_t](#) [unpack\\_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, [unpack\\_swiloc\\_SwiLocSetAutoStart\\_t](#) \*pOutput)

### 9.37.1 Typedef Documentation

9.37.1.1 typedef [unpack\\_result\\_t](#) [unpack\\_swiloc\\_SwiLocSetAutoStart\\_t](#)

## 9.37.2 Function Documentation

### 9.37.2.1 `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

### 9.37.2.2 `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
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

### 9.37.2.3 `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

9.37.2.4 `int unpack_swiloc_SwiLocSetAutoStart ( uint8_t * pResp, uint16_t respLen, unpack_swiloc_SwiLocSetAutoStart_t * pOutput )`

Set 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

## 9.38 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 LITEQMI_MAX_SWIOMA_STR_LEN 255`

### Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMCancelSession\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMSetSettings\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMSelectSelection\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMAAlertCallback\\_t](#)

## Functions

- int [pack\\_swima\\_SLQSOMADMStartSession](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swima\\_SLQSOMADMStartSession\\_t](#) reqParam)
- int [unpack\\_swima\\_SLQSOMADMStartSession](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swima\\_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, [unpack\\_swima\\_SLQSOMADMCancelSession\\_t](#) \*pOutput)
- 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, [unpack\\_swima\\_SLQSOMADMSetSettings\\_t](#) \*pOutput)
- 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, [unpack\\_swima\\_SLQSOMADMSendSelection\\_t](#) \*pOutput)
- 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, [unpack\\_swima\\_SLQSOMADMAAlertCallback\\_t](#) \*pOutput)
- int [unpack\\_swima\\_SLQSOMADMAAlertCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swima\\_SLQSOMADMAAlertCallback\\_ind\\_t](#) \*pOutput)

### 9.38.1 Macro Definition Documentation

9.38.1.1 `#define LITEQMI_MAX_SWIOMA_STR_LEN 255`

### 9.38.2 Typedef Documentation

9.38.2.1 `typedef unpack_result_t unpack_swima_SLQSOMADMAAlertCallback_t`

9.38.2.2 `typedef unpack_result_t unpack_swima_SLQSOMADMCancelSession_t`

9.38.2.3 `typedef unpack_result_t unpack_swima_SLQSOMADMSendSelection_t`

9.38.2.4 `typedef unpack_result_t unpack_swima_SLQSOMADMSetSettings_t`

### 9.38.3 Function Documentation

9.38.3.1 `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

**9.38.3.2** `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

9.38.3.3 `int pack_swima_SLQSOMADMGetSessionInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_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

9.38.3.4 `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

**9.38.3.5** `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

9.38.3.6 `int pack_swima_SLQSOMADMSetSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_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_swima_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

9.38.3.7 `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

**9.38.3.8** `int unpack_swioma_SLQSOMADMAAlertCallback ( uint8_t * pResp, uint16_t respLen, unpack_swioma_SLQSOMADMAAlertCallback_t * pOutput )`

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] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem
<i>pOutput</i>	[out] • response unpacked

## 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

**9.38.3.9** `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] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem

<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

9.38.3.10 int unpack\_swioma\_SLQSOMADMCancelSession ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swioma\_SLQSOMADMCancelSession\_t \* *pOutput* )

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>
<i>pOutput</i>	[out] <ul style="list-style-type: none"> <li>response unpacked</li> </ul>

**Returns**

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

**See Also**

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

9.38.3.11 int unpack\_swioma\_SLQSOMADMGetSessionInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swioma\_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_swioma_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

**9.38.3.12** int unpack\_swioma\_SLQSOMADMGetSettings ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swioma\_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

**9.38.3.13** int unpack\_swioma\_SLQSOMADMSendSelection ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swioma\_SLQSOMADMSendSelection\_t \* *pOutput* )

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>
<i>pOutput</i>	[out] <ul style="list-style-type: none"> <li>response unpacked</li> </ul>

## Returns

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

## See Also

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

9.38.3.14 `int unpack_swima_SLQSOMADMSetSettings ( uint8_t * pResp, uint16_t respLen,  
unpack_swima_SLQSOMADMSetSettings_t * pOutput )`

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettings3

## Parameters

<i>pResp</i>	[in] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem
<i>pOutput</i>	[out] • response unpacked

## Returns

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

## See Also

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

9.38.3.15 `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] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem
<i>pOutput</i>	[out] • See <a href="#">unpack_swima_SLQSOMADMStartSession_t</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

## 9.39 swiomaext.h File Reference

### Data Structures

- struct [pack\\_swioma\\_SLQSOMADMStartSessionExt\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMCancelSessionExt\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMSetSettingsExt\\_t](#)
- struct [pack\\_swioma\\_SLQSOMADMSendSelectionExt\\_t](#)
- struct [unpack\\_swioma\\_SLQSOMADMGetSessionInfoExt\\_t](#)

### Macros

- `#define LITEQMI_MAX_SWIOMA_STR_LEN 255`
- `#define LITE_SWIOMAEXT_MAX_UCS2_DATA_LEN 512`

### Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMStartSessionExt\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMCancelSessionExt\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMSetSettingsExt\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_swioma\\_SLQSOMADMSendSelectionExt\\_t](#)

### Functions

- int [pack\\_swioma\\_SLQSOMADMStartSessionExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swioma\\_SLQSOMADMStartSessionExt\\_t](#) reqParam)
- int [unpack\\_swioma\\_SLQSOMADMStartSessionExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMStartSessionExt\\_t](#) \*pOutput)
- int [pack\\_swioma\\_SLQSOMADMCancelSessionExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swioma\\_SLQSOMADMCancelSessionExt\\_t](#) reqParam)
- int [unpack\\_swioma\\_SLQSOMADMCancelSessionExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMCancelSessionExt\\_t](#) \*pOutput)
- int [pack\\_swioma\\_SLQSOMADMSetSettingsExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swioma\\_SLQSOMADMSetSettingsExt\\_t](#) reqParam)
- int [unpack\\_swioma\\_SLQSOMADMSetSettingsExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMSetSettingsExt\\_t](#) \*pOutput)
- int [pack\\_swioma\\_SLQSOMADMSendSelectionExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_swioma\\_SLQSOMADMSendSelectionExt\\_t](#) reqParam)
- int [unpack\\_swioma\\_SLQSOMADMSendSelectionExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMSendSelectionExt\\_t](#) \*pOutput)
- int [pack\\_swioma\\_SLQSOMADMGetSessionInfoExt](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_swioma\\_SLQSOMADMGetSessionInfoExt](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_swioma\\_SLQSOMADMGetSessionInfoExt\\_t](#) \*pOutput)

#### 9.39.1 Macro Definition Documentation

9.39.1.1 `#define LITE_SWIOMAEXT_MAX_UCS2_DATA_LEN 512`

9.39.1.2 `#define LITEQMI_MAX_SWIOMA_STR_LEN 255`

#### 9.39.2 Typedef Documentation

9.39.2.1 `typedef unpack_result_t unpack_swima_SLQSOMADMCancelSessionExt_t`

9.39.2.2 `typedef unpack_result_t unpack_swima_SLQSOMADMSendSelectionExt_t`

9.39.2.3 `typedef unpack_result_t unpack_swima_SLQSOMADMSetSettingsExt_t`

9.39.2.4 `typedef unpack_result_t unpack_swima_SLQSOMADMStartSessionExt_t`

### 9.39.3 Function Documentation

9.39.3.1 `int pack_swima_SLQSOMADMCancelSessionExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMCancelSessionExt_t reqParam )`

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSessionExt

#### Parameters

<i>pCtx</i>	[OUT] • See <a href="#">pack_qmi_t</a> for more information
<i>pReqBuf</i>	[IN/OUT] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
<i>pLen</i>	[IN/OUT] • On input, size of pReqBuf • On output, number of bytes actually packed
<i>reqParam</i>	[IN] • See <a href="#">pack_swima_SLQSOMADMCancelSessionExt_t</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

#### Note

Timeout: 20 seconds

9.39.3.2 `int pack_swima_SLQSOMADMGetSessionInfoExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Function to pack OMA-DM session info command This maps to SLQSOMADMGetSessionInfoExt

#### Parameters

<i>pCtx</i>	[OUT] • See <a href="#">pack_qmi_t</a> for more information
<i>pReqBuf</i>	[IN/OUT] • Buffer for packed QMI command to be provided by the host application • Minimum expected size is 2048 bytes
Generated on Fri Feb 14 2020 02:35:32 for LinuxQMISDK-Lite by Doxygen	

<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

**9.39.3.3** `int pack_swioma_SLQSOMADMSendSelectionExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swioma_SLQSOMADMSendSelectionExt_t reqParam )`

Function to pack OMA-DM send selection command This maps to SLQSOMADMSendSelectionExt

**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_SLQSOMADMSendSelectionExt_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

9.39.3.4 `int pack_swima_SLQSOMADMSetSettingsExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMSetSettingsExt_t reqParam )`

Function to pack OMA-DM set settings command This maps to SLQSOMADMSetSettingsExt

#### 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_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

9.39.3.5 `int pack_swima_SLQSOMADMStartSessionExt ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_swima_SLQSOMADMStartSessionExt_t reqParam )`

Function to pack Start OMA-DM session command This maps to SLQSOMADMStartSessionExt

#### 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_SLQSOMADMStartSessionExt_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

**9.39.3.6** `int unpack_swioma_SLQSOMADMCancelSessionExt ( uint8_t * pResp, uint16_t respLen, unpack_swioma_SLQSOMADMCancelSessionExt_t * pOutput )`

Function to pack cancel OMA-DM session command This maps to SLQSOMADMCancelSessionExt

## Parameters

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • response unpacked

## Returns

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

## See Also

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

**9.39.3.7** `int unpack_swioma_SLQSOMADMGetSessionInfoExt ( uint8_t * pResp, uint16_t respLen, unpack_swioma_SLQSOMADMGetSessionInfoExt_t * pOutput )`

Function to unpack information related to the current (or previous if no session is active) OMA-DM session. This maps to SLQSOMADMGetSessionInfoExt

## Parameters

<i>pResp</i>	[in] • Response from modem
<i>respLen</i>	[in] • Length of pResp from modem
<i>pOutput</i>	[out] • See <a href="#">unpack_swioma_SLQSOMADMGetSessionInfoExt_t</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

**9.39.3.8** int unpack\_swima\_SLQSOMADMSendSelectionExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swima\_SLQSOMADMSendSelectionExt\_t \* *pOutput* )

Function to unpack OMA-DM send selection command This maps to SLQSOMADMSendSelectionExt

**Parameters**

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • response unpacked

**Returns**

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

**See Also**

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

**9.39.3.9** int unpack\_swima\_SLQSOMADMSetSettingsExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swima\_SLQSOMADMSetSettingsExt\_t \* *pOutput* )

Function to unpack OMA-DM set settings command This maps to SLQSOMADMSetSettingsExt

**Parameters**

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • response unpacked

**Returns**

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

## See Also

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

9.39.3.10 int unpack\_swioma\_SLQSOMADMStartSessionExt ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_swioma\_SLQSOMADMStartSessionExt\_t \* *pOutput* )

Function to unpack Start OMA-DM session response from modem This maps to SLQSOMADMStartSessionExt

## Parameters

<i>pResp</i>	[IN] • Response from modem
<i>respLen</i>	[IN] • Length of pResp from modem
<i>pOutput</i>	[OUT] • response unpacked

## Returns

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

## See Also

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

## 9.40 switype\_256bit.h File Reference

### Data Structures

- struct [swi\\_uint256\\_t](#)

### Macros

- #define [SWI\\_UINT256\\_WORD\\_COUNT](#) 16
- #define [SWI\\_UINT256\\_BITS\\_PER\\_WORD](#) 16
- #define [SWI\\_UINT256\\_WORD\\_OFFSET](#)(val) ((val) / [SWI\\_UINT256\\_BITS\\_PER\\_WORD](#))
- #define [SWI\\_UINT256\\_BIT\\_OFFSET](#)(val) ((val) % [SWI\\_UINT256\\_BITS\\_PER\\_WORD](#))
- #define [SWI\\_UINT256\\_INT\\_VALUE](#) {{0}}

### Typedefs

- typedef void(\* [logger](#) )(uint8\_t lvl, const char \*buff)

### Functions

- void [swi\\_uint256\\_set\\_bit](#) ([swi\\_uint256\\_t](#) \*pMask, uint8\_t bit)
- void [swi\\_uint256\\_clear\\_bit](#) ([swi\\_uint256\\_t](#) \*pMask, uint8\_t bit)
- uint8\_t [swi\\_uint256\\_get\\_bit](#) ([swi\\_uint256\\_t](#) Mask, uint8\_t bit)
- void [swi\\_uint256\\_print\\_mask](#) ([swi\\_uint256\\_t](#) mask)

### 9.40.1 Macro Definition Documentation

9.40.1.1 `#define SWI_UINT256_BIT_OFFSET( val ) ((val) % SWI_UINT256_BITS_PER_WORD)`

9.40.1.2 `#define SWI_UINT256_BITS_PER_WORD 16`

9.40.1.3 `#define SWI_UINT256_INT_VALUE {{0}}`

9.40.1.4 `#define SWI_UINT256_WORD_COUNT 16`

9.40.1.5 `#define SWI_UINT256_WORD_OFFSET( val ) ((val) / SWI_UINT256_BITS_PER_WORD)`

### 9.40.2 Typedef Documentation

9.40.2.1 `typedef void(* logger)(uint8_t lvl, const char *buff)`

### 9.40.3 Function Documentation

9.40.3.1 `void swi_uint256_clear_bit ( swi_uint256_t * pMask, uint8_t bit )`

Clear relevant bit in provided mask

#### Parameters

in	<i>pMask</i>	Pointer to bitmask
in	<i>bit</i>	Bit to clear

9.40.3.2 `uint8_t swi_uint256_get_bit ( swi_uint256_t Mask, uint8_t bit )`

Get relevant bit from provided mask

#### Parameters

in	<i>Mask</i>	Bit mask
in	<i>bit</i>	Bit to extract

#### Returns

Bit value

- 0 - bit is not set
- 1 - bit is set

9.40.3.3 `void swi_uint256_print_mask ( swi_uint256_t mask )`

Print the log mask to syslog. Use only for debugging

#### Parameters

in	<i>mask</i>	Bit mask
----	-------------	----------

9.40.3.4 `void swi_uint256_set_bit ( swi_uint256_t * pMask, uint8_t bit )`

Set relevant bit in provided mask

## Parameters

in	<i>pMask</i>	Pointer to bit mask
in	<i>bit</i>	Bit to set

## 9.41 SWIWWANCMAPI.h File Reference

## 9.42 tmd.h File Reference

## Data Structures

- struct [tmd\\_mitigationDevList](#)
- struct [unpack\\_tmd\\_SLQSTmdGetMitigationDevList\\_t](#)
- struct [pack\\_tmd\\_SLQSTmdGetMitigationLvl\\_t](#)
- struct [unpack\\_tmd\\_SLQSTmdGetMitigationLvl\\_t](#)
- struct [pack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t](#)
- struct [unpack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t](#)
- struct [pack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl\\_t](#)
- struct [unpack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl\\_t](#)
- struct [unpack\\_tmd\\_SLQSTmdMitigationLvlRptCallback\\_ind\\_t](#)

## Macros

- `#define TMD_MAX_DEV_LIST 255`
- `#define MAX_MITIGATION_DEV_LIST_LEN 255`
- `#define MAX_MITIGATION_DEV_ID_LEN 255`

## Functions

- int [pack\\_tmd\\_SLQSTmdGetMitigationDevList](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_tmd\\_SLQSTmdGetMitigationDevList](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_tmd\\_SLQSTmdGetMitigationDevList\\_t](#) \*pOutput)
- int [pack\\_tmd\\_SLQSTmdGetMitigationLvl](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_tmd\\_SLQSTmdGetMitigationLvl\\_t](#) \*reqArg)
- int [unpack\\_tmd\\_SLQSTmdGetMitigationLvl](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_tmd\\_SLQSTmdGetMitigationLvl\\_t](#) \*pOutput)
- int [pack\\_tmd\\_SLQSTmdRegNotMitigationLvl](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t](#) \*reqArg)
- int [unpack\\_tmd\\_SLQSTmdRegNotMitigationLvl](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_tmd\\_SLQSTmdRegNotMitigationLvl\\_t](#) \*pOutput)
- int [pack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl\\_t](#) \*reqArg)
- int [unpack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_tmd\\_SLQSTmdDeRegNotMitigationLvl\\_t](#) \*pOutput)
- int [unpack\\_tmd\\_SLQSTmdMitigationLvlRptCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_tmd\\_SLQSTmdMitigationLvlRptCallback\\_ind\\_t](#) \*pOutput)

## 9.42.1 Macro Definition Documentation

9.42.1.1 `#define MAX_MITIGATION_DEV_ID_LEN 255`

9.42.1.2 `#define MAX_MITIGATION_DEV_LIST_LEN 255`

9.42.1.3 `#define TMD_MAX_DEV_LIST 255`

## 9.42.2 Function Documentation

9.42.2.1 `int pack_tmd_SLQSTmdDeRegNotMitigationLvl ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_tmd_SLQSTmdDeRegNotMitigationLvl_t * reqArg )`

To DeRegister notification mitigation level pack

### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqArg</i>	request argument.

### Returns

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

### See Also

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

9.42.2.2 `int pack_tmd_SLQSTmdGetMitigationDevList ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

To get mitigation device 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

### Returns

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

### See Also

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

9.42.2.3 `int pack_tmd_SLQSTmdGetMitigationLvl ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_tmd_SLQSTmdGetMitigationLvl_t * reqArg )`

To get mitigation level pack

### Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqArg</i>	request argument.

## Returns

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

## See Also

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

9.42.2.4 `int pack_tmd_SLQSTmdRegNotMitigationLvl ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_tmd_SLQSTmdRegNotMitigationLvl_t * reqArg )`

To Register notification mitigation level pack

## Parameters

in, out	<i>pCtx</i>	qmi request context
out	<i>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
out	<i>reqArg</i>	request argument.

## Returns

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

## See Also

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

9.42.2.5 `int unpack_tmd_SLQSTmdDeRegNotMitigationLvl ( uint8_t * pResp, uint16_t respLen, unpack_tmd_SLQSTmdDeRegNotMitigationLvl_t * pOutput )`

To De-Register notification mitigation level 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

9.42.2.6 `int unpack_tmd_SLQSTmdGetMitigationDevList ( uint8_t * pResp, uint16_t respLen, unpack_tmd_SLQSTmdGetMitigationDevList_t * pOutput )`

To mitigation device 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

9.42.2.7 int unpack\_tmd\_SLQSTmdGetMitigationLvl ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_tmd\_SLQSTmdGetMitigationLvl\_t \* *pOutput* )

To get mitigation level 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

9.42.2.8 int unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t \* *pOutput* )

Mitigation Level Report Indication unpack

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response 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



9.42.2.9 `int unpack_tmd_SLQSTmdRegNotMitigationLvl ( uint8_t * pResp, uint16_t respLen, unpack_tmd_SLQSTmdRegNotMitigationLvl_t * pOutput )`

To Register notification mitigation level 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

## 9.43 uim.h File Reference

### Data Structures

- struct [uim\\_appStatus](#)
- struct [uim\\_slotInfo](#)
- struct [uim\\_cardStatus](#)
- struct [uim\\_hotSwapStatus](#)
- struct [uim\\_validCardStatus](#)
- struct [uim\\_simBusyStatus](#)
- struct [unpack\\_uim\\_GetCardStatus\\_t](#)
- struct [unpack\\_uim\\_GetCardStatusV2\\_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 [unpack\\_uim\\_UnblockPinV2\\_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\\_SLQSUIEventRegister\\_t](#)
- struct [unpack\\_uim\\_SLQSUIEventRegister\\_t](#)
- struct [appStats](#)
- struct [slotInf](#)
- struct [unpack\\_uim\\_SLQSUISetStatusChangeCallBack\\_ind\\_t](#)
- struct [slot\\_t](#)
- struct [slots\\_t](#)
- struct [uim\\_physlotInfo](#)
- struct [uim\\_physlotsInfo](#)
- struct [unpack\\_uim\\_SLQSUIGetSlotsStatus\\_t](#)
- struct [uim\\_GetSlotsStatusTlv](#)
- struct [uim\\_GetSlotsInfoTlv](#)
- struct [unpack\\_uim\\_SLQSUIGetSlotsStatusV2\\_t](#)
- struct [pack\\_uim\\_SLQSUISwitchSlot\\_t](#)
- struct [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind\\_t](#)
- struct [pack\\_uim\\_SLQSUIPowerUp\\_t](#)
- struct [pack\\_uim\\_SLQSUIPowerDown\\_t](#)
- struct [pack\\_uim\\_SLQSUIRefreshOK\\_t](#)
- struct [uim\\_registerRefresh](#)
- struct [pack\\_uim\\_SLQSUIRefreshRegister\\_t](#)
- struct [pack\\_uim\\_SLQSUIRefreshComplete\\_t](#)
- struct [uim\\_refreshevent](#)
- struct [pack\\_uim\\_SLQSUIRefreshGetLastEvent\\_t](#)
- struct [unpack\\_uim\\_SLQSUIRefreshGetLastEvent\\_t](#)
- struct [pack\\_uim\\_SLQSUIGetFileAttributes\\_t](#)
- struct [uim\\_fileAttributes](#)
- struct [unpack\\_uim\\_SLQSUIGetFileAttributes\\_t](#)
- struct [uim\\_depersonalizationInformation](#)
- struct [pack\\_uim\\_SLQSUIDepersonalization\\_t](#)
- struct [unpack\\_uim\\_SLQSUIDepersonalization\\_t](#)
- struct [uim\\_authenticationData](#)
- struct [pack\\_uim\\_SLQSUIAuthenticate\\_t](#)
- struct [uim\\_authenticateResult](#)
- struct [unpack\\_uim\\_SLQSUIAuthenticate\\_t](#)
- struct [uim\\_personalizationStatus](#)
- struct [pack\\_uim\\_SLQSUIGetConfiguration\\_t](#)
- struct [unpack\\_uim\\_SLQSUIGetConfiguration\\_t](#)
- struct [unpack\\_uim\\_SLQSUIRefreshCallback\\_Ind\\_t](#)
- struct [pack\\_uim\\_SLQSUISetServiceStatus\\_t](#)
- struct [pack\\_uim\\_SLQSUIGetServiceStatus\\_t](#)
- struct [uim\\_UIMGetFDNStatus](#)
- struct [uim\\_UIMGetHiddenKeyStatus](#)
- struct [uim\\_UIMGetIndex](#)
- struct [unpack\\_uim\\_SLQSUIGetServiceStatus\\_t](#)
- struct [uim\\_readRecordInfo](#)
- struct [pack\\_uim\\_SLQSUIReadRecord\\_t](#)
- struct [uim\\_cardResultInfo](#)
- struct [uim\\_readResultInfo](#)
- struct [uim\\_additionalReadResult](#)
- struct [uim\\_indToken](#)
- struct [unpack\\_uim\\_SLQSUIReadRecord\\_t](#)
- struct [uim\\_writeRecordInfo](#)
- struct [pack\\_uim\\_SLQSUIWriteRecord\\_t](#)
- struct [unpack\\_uim\\_SLQSUIWriteRecord\\_t](#)
- struct [uim\\_writeTransparentInfo](#)
- struct [pack\\_uim\\_SLQSUIWriteTransparent\\_t](#)
- struct [unpack\\_uim\\_SLQSUIWriteTransparent\\_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`
- `#define UIM_MAX_CONTENT_LENGTH 1024`
- `#define UIM_MAX_ACTIVE_PERS_FEATURES 12`
- `#define MAX_ATR_LENGTH 255`
- `#define MAX_PHY_SLOTS_INFO 255`

## Typedefs

- `typedef unpack_result_t unpack_uim_SLQSUIMSwitchSlot_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMPowerUp_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMPowerDown_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMReset_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMRefreshOK_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMRefreshRegister_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMRefreshComplete_t`
- `typedef unpack_result_t unpack_uim_SLQSUIMSetServiceStatus_t`

## 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 unpack_uim_GetCardStatusV2 (uint8_t *pResp, uint16_t respLen, unpack_uim_GetCardStatusV2_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 unpack_uim_UnblockPinV2 (uint8_t *pResp, uint16_t respLen, unpack_uim_UnblockPinV2_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 unpack\\_uim\\_SLQSUIMGetSlotsStatusV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetSlotsStatusV2\\_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, [unpack\\_uim\\_SLQSUIMSwitchSlot\\_t](#) \*pOutput)
- [int unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SetUimSlotStatusChangeCallback\\_ind\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMPowerUp](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMPowerUp\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMPowerUp](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMPowerUp\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMPowerDown](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMPowerDown\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMPowerDown](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMPowerDown\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMReset](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_uim\\_SLQSUIMReset](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMReset\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMRefreshOK](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMRefreshOK\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMRefreshOK](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMRefreshOK\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMRefreshRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMRefreshRegister\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMRefreshRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMRefreshRegister\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMRefreshComplete](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMRefreshComplete\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMRefreshComplete](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMRefreshComplete\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMRefreshGetLastEvent](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMRefreshGetLastEvent\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMRefreshGetLastEvent](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMRefreshGetLastEvent\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMGetFileAttributes](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMGetFileAttributes\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMGetFileAttributes](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetFileAttributes\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMDepersonalization](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMDepersonalization\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMDepersonalization](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMDepersonalization\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMAuthenticate](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMAuthenticate\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMAuthenticate](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMAuthenticate\\_t](#) \*pOutput)
- [int pack\\_uim\\_SLQSUIMGetConfiguration](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMGetConfiguration\\_t](#) \*reqArg)
- [int unpack\\_uim\\_SLQSUIMGetConfiguration](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetConfiguration\\_t](#) \*pOutput)

- int [unpack\\_uim\\_SLQSUIMRefreshCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMRefreshCallback\\_Ind\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMSetServiceStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMSetServiceStatus\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMSetServiceStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMSetServiceStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMGetServiceStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMGetServiceStatus\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMGetServiceStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMGetServiceStatus\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMReadRecord](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMReadRecord\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMReadRecord](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMReadRecord\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMWriteRecord](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMWriteRecord\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMWriteRecord](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMWriteRecord\\_t](#) \*pOutput)
- int [pack\\_uim\\_SLQSUIMWriteTransparent](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_uim\\_SLQSUIMWriteTransparent\\_t](#) \*reqArg)
- int [unpack\\_uim\\_SLQSUIMWriteTransparent](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_uim\\_SLQSUIMWriteTransparent\\_t](#) \*pOutput)

### 9.43.1 Macro Definition Documentation

9.43.1.1 `#define MAX_ATR_LENGTH 255`

9.43.1.2 `#define MAX_DESCRIPTION_LENGTH 255`

9.43.1.3 `#define MAX_ICCID_LENGTH 255`

9.43.1.4 `#define MAX_NO_OF_APPLICATIONS 10`

9.43.1.5 `#define MAX_NO_OF_SLOTS 5`

9.43.1.6 `#define MAX_PHY_SLOTS_INFO 255`

9.43.1.7 `#define MAX_SLOTS_STATUS 255`

9.43.1.8 `#define UIM_MAX_ACTIVE_PERS_FEATURES 12`

9.43.1.9 `#define UIM_MAX_CONTENT_LENGTH 1024`

9.43.1.10 `#define UIM_MAX_DESCRIPTION_LENGTH 255`

9.43.1.11 `#define UIM_MAX_NO_OF_APPLICATIONS 10`

9.43.1.12 `#define UIM_MAX_NO_OF_SLOTS 5`

9.43.1.13 `#define UIM_UINT8_MAX_STRING_SZ 255`

### 9.43.2 Typedef Documentation

9.43.2.1 `typedef unpack_result_t unpack_uim_SLQSUIMPowerDown_t`

9.43.2.2 `typedef unpack_result_t unpack_uim_SLQSUIMPowerUp_t`

9.43.2.3 `typedef unpack_result_t unpack_uim_SLQSUIMRefreshComplete_t`

9.43.2.4 `typedef unpack_result_t unpack_uim_SLQSUIMRefreshOK_t`

9.43.2.5 `typedef unpack_result_t unpack_uim_SLQSUIMRefreshRegister_t`

9.43.2.6 `typedef unpack_result_t unpack_uim_SLQSUIMReset_t`

9.43.2.7 `typedef unpack_result_t unpack_uim_SLQSUIMSetServiceStatus_t`

9.43.2.8 `typedef unpack_result_t unpack_uim_SLQSUIMSwitchSlot_t`

### 9.43.3 Function Documentation

9.43.3.1 `int pack_uim_ChangePin ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_ChangePin_t * reqArg )`

Change Pin pack

#### Parameters

in, 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

9.43.3.2 `int pack_uim_GetCardStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, void * reqArg )`

Get Card 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
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

9.43.3.3 `int pack_uim_ReadTransparent ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_ReadTransparent_t * reqArg )`

SLQS ReadTransparent pack

#### Parameters

in, 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

9.43.3.4 `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, 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

9.43.3.5 `int pack_uim_SLQSUIMAuthenticate ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIMAuthenticate_t * reqArg )`

Send a security command to the card 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

**9.43.3.6** `int pack_uim_SLQSUIDepersonalization ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIDepersonalization_t * reqArg )`

De-activates or unblocks the personalization on the phone 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

**9.43.3.7** `int pack_uim_SLQSUIEventRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIEventRegister_t * reqArg )`

UIM Status Change callback enable 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

**9.43.3.8** `int pack_uim_SLQSUIGetConfiguration ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIGetConfiguration_t * reqArg )`

Gets the modem configuration for the UIM module 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

9.43.3.9 int pack\_uim\_SLQSUIMGetFileAttributes ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_uim\_SLQSUIMGetFileAttributes\_t \* *reqArg* )

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path 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

9.43.3.10 int pack\_uim\_SLQSUIMGetServiceStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_uim\_SLQSUIMGetServiceStatus\_t \* *reqArg* )

Pack Get Service Status.

## 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

9.43.3.11 `int pack_uim_SLQSUIGetSlotsStatus ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Pack get slots status.

#### 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

9.43.3.12 `int pack_uim_SLQSUIPowerDown ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIPowerDown_t * reqArg )`

Powers down the card 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

9.43.3.13 `int pack_uim_SLQSUIPowerUp ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIPowerUp_t * reqArg )`

Powers up the card 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

9.43.3.14 `int pack_uim_SLQSUIReadRecord ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIReadRecord_t * reqArg )`

Pack Read Record Status

## 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

9.43.3.15 `int pack_uim_SLQSUIRefreshComplete ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIRefreshComplete_t * reqArg )`

Enables the terminal response to be sent to the card When the client finished Refresh procedure 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

9.43.3.16 `int pack_uim_SLQSUIRefreshGetLastEvent ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIRefreshGetLastEvent_t * reqArg )`

Retrieve the last refresh event 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

**9.43.3.17** int pack\_uim\_SLQSUIRefreshOK ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_uim\_SLQSUIRefreshOK\_t \* reqArg )

Enables the client to indicate whether it is OK to start the Refresh procedure pack.

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	reqArg	request parameter

**Returns**

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

**See Also**

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

**9.43.3.18** int pack\_uim\_SLQSUIRefreshRegister ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_uim\_SLQSUIRefreshRegister\_t \* reqArg )

Registers for file change notifications triggered by the card pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	reqArg	request parameter

**Returns**

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

**See Also**

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

**9.43.3.19** int pack\_uim\_SLQSUIReset ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Resets the issuing control points state kept by the service 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

9.43.3.20 int pack\_uim\_SLQSUI/SetServiceStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_uim\_SLQSUI/SetServiceStatus\_t \* *reqArg* )

Pack Set Service Status.

## 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

9.43.3.21 int pack\_uim\_SLQSUI/SwitchSlot ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_uim\_SLQSUI/SwitchSlot\_t \* *reqArg* )

switch slot 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

9.43.3.22 `int pack_uim_SLQSUIWriteRecord ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIWriteRecord_t * reqArg )`

Pack Write Record data

#### 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

9.43.3.23 `int pack_uim_SLQSUIWriteTransparent ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_SLQSUIWriteTransparent_t * reqArg )`

Pack Write Transparent

#### 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

9.43.3.24 `int pack_uim_UnblockPin ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_uim_UnblockPin_t * reqArg )`

Unblock Pin 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

9.43.3.25 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, 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

9.43.3.26 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

9.43.3.27 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

**9.43.3.28** int unpack\_uim\_GetCardStatusV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_GetCardStatusV2\_t \* *pOutput* )

Get Card Status unpack V2

**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

**9.43.3.29** 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

**9.43.3.30** 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

**9.43.3.31** 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

**9.43.3.32** int unpack\_uim\_SLQSUIAuthenticate ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUI-  
Authenticate\_t \* *pOutput* )

Unpack the status code received from the card when card responded to the read request

**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

**9.43.3.33** int unpack\_uim\_SLQSUIDepersonalization ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIDepersonalization\_t \* *pOutput* )

De-activates or unblocks the personalization on the phone 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

**9.43.3.34** int unpack\_uim\_SLQSUIEventRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIEventRegister\_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

**9.43.3.35** int unpack\_uim\_SLQSUIGetConfiguration ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIGetConfiguration\_t \* *pOutput* )

Gets the modem configuration for the UIM module 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

9.43.3.36 `int unpack_uim_SLQSUIMGetFileAttributes ( uint8_t * pResp, uint16_t respLen,  
unpack_uim_SLQSUIMGetFileAttributes_t * pOutput )`

Retrieves the file attributes for any EF(Elementary File) or DF(Dedicated File) in the card and provides access by the path 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

9.43.3.37 `int unpack_uim_SLQSUIMGetServiceStatus ( uint8_t * pResp, uint16_t respLen,  
unpack_uim_SLQSUIMGetServiceStatus_t * pOutput )`

Get Service 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

9.43.3.38 `int unpack_uim_SLQSUIMGetSlotsStatus ( uint8_t * pResp, uint16_t respLen,  
unpack_uim_SLQSUIMGetSlotsStatus_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

9.43.3.39 `int unpack_uim_SLQSUIGetSlotsStatusV2 ( uint8_t * pResp, uint16_t respLen,  
unpack_uim_SLQSUIGetSlotsStatusV2_t * pOutput )`

get slot status unpack V2

## 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

9.43.3.40 `int unpack_uim_SLQSUIPowerDown ( uint8_t * pResp, uint16_t respLen, unpack_uim_SLQSUIPower-  
Down_t * pOutput )`

Powers down the card 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

9.43.3.41 `int unpack_uim_SLQSUIPowerUp ( uint8_t * pResp, uint16_t respLen, unpack_uim_SLQSUIPowerUp_t *  
pOutput )`

Powers up the card 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

**9.43.3.42** int unpack\_uim\_SLQSUIReadRecord ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIReadRecord\_t \* *pOutput* )

Read Record 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

**9.43.3.43** int unpack\_uim\_SLQSUIRefreshCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIRefreshCallback\_Ind\_t \* *pOutput* )

Unpack UIM refresh event indication.

## 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

**9.43.3.44** int unpack\_uim\_SLQSUIRefreshComplete ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIRefreshComplete\_t \* *pOutput* )

Enables the terminal response to be sent to the card When the client finished Refresh procedure 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

**9.43.3.45** int unpack\_uim\_SLQSUIRefreshGetLastEvent ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIRefreshGetLastEvent\_t \* *pOutput* )

Retrieve the last refresh event 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

**9.43.3.46** int unpack\_uim\_SLQSUIRefreshOK ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIRefreshOK\_t  
\* *pOutput* )

Enables the client to indicate whether it is OK to start the Refresh procedure 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

**9.43.3.47** int unpack\_uim\_SLQSUIRefreshRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_uim\_SLQSUIRefreshRegister\_t \* *pOutput* )

Registers for file change notifications triggered by the card 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

9.43.3.48 int unpack\_uim\_SLQSUIReset ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIReset\_t \* *pOutput* )

Resets the issuing control points state kept by the service unpack.

## Parameters

in	<i>pResp</i>	qmi response
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

9.43.3.49 int unpack\_uim\_SLQSUISetServiceStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUISetServiceStatus\_t \* *pOutput* )

Set Service 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

9.43.3.50 int unpack\_uim\_SLQSUISetStatusChangeCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUISetStatusChangeCallback\_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\_SLQSUIEventRegister to subscribe

9.43.3.51 int unpack\_uim\_SLQSUISwitchSlot ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUISwitchSlot\_t \* *pOutput* )

switch slot 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

9.43.3.52 int unpack\_uim\_SLQSUIWriteRecord ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIWriteRecord\_t \* *pOutput* )

Write Record 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

9.43.3.53 int unpack\_uim\_SLQSUIWriteTransparent ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_SLQSUIWriteTransparent\_t \* *pOutput* )

Write Transparent 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

9.43.3.54 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

9.43.3.55 int unpack\_uim\_UnblockPinV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_uim\_UnblockPinV2\_t \* *pOutput* )

Unblock Pin unpack V2

## 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

9.43.3.56 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

## 9.44 voice.h File Reference

## Data Structures

- struct [voice\\_USSInfo](#)
- struct [pack\\_voice\\_OriginateUSSD\\_t](#)
- struct [pack\\_voice\\_AnswerUSSD\\_t](#)
- struct [voice\\_UUSInfo](#)
- struct [voice\\_CUGInfo](#)
- struct [voice\\_calledPartySubAdd](#)
- struct [pack\\_voice\\_SLQSVoiceDialCall\\_t](#)
- struct [voice\\_alphaIDInfo](#)
- struct [voice\\_ccSUPSType](#)
- struct [unpack\\_voice\\_SLQSVoiceDialCall\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceEndCall\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceEndCall\\_t](#)
- struct [voice\\_callFwdTypeAndPlan](#)
- struct [pack\\_voice\\_SLQSVoiceSetSUPSService\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceSetSUPSService\\_t](#)
- struct [voice\\_airTimer](#)
- struct [voice\\_roamTimer](#)
- struct [voice\\_prefVoiceSO](#)
- struct [pack\\_voice\\_SLQSVoiceSetConfig\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceSetConfig\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceAnswerCall\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceAnswerCall\\_t](#)
- struct [voice\\_CLIRResp](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCLIR\\_t](#)
- struct [voice\\_CLIPResp](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCLIP\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceGetCallWaiting\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCallWaiting\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceGetCallBarring\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCallBarring\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceGetCallForwardingStatus\\_t](#)
- struct [voice\\_callFWInfo](#)
- struct [voice\\_callFWExtInfo](#)
- struct [voice\\_getCallFWInfo](#)
- struct [voice\\_getCallFWExtInfo](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCallForwardingStatus\\_t](#)

- struct [pack\\_voice\\_SLQSVoiceSetCallBarringPassword\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceSetCallBarringPassword\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceGetCallInfo\\_t](#)
- struct [voice\\_callInfo](#)
- struct [voice\\_remotePartyNum](#)
- struct [voice\\_remotePartyName](#)
- struct [voice\\_connectNumInfo](#)
- struct [voice\\_diagInfo](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCallInfo\\_t](#)
- struct [voice\\_getAllCallInformation](#)
- struct [voice\\_getAllCallRmtPtyNum](#)
- struct [voice\\_getAllCallRmtPtyName](#)
- struct [voice\\_allCallsUUSInfo](#)
- struct [voice\\_allCallsAlphaIDInfo](#)
- struct [voice\\_allCallsDiagInfo](#)
- struct [voice\\_peerNumberInfo](#)
- struct [voice\\_arrCallInfo](#)
- struct [voice\\_arrRemotePartyNum](#)
- struct [voice\\_arrRemotePartyName](#)
- struct [voice\\_arrAlertingType](#)
- struct [voice\\_arrUUSInfo](#)
- struct [voice\\_arrSvcOption](#)
- struct [voice\\_arrCallEndReason](#)
- struct [voice\\_arrAlphaID](#)
- struct [voice\\_arrConnectPartyNum](#)
- struct [voice\\_arrDiagInfo](#)
- struct [voice\\_arrCalledPartyNum](#)
- struct [voice\\_arrRedirPartyNum](#)
- struct [voice\\_arrAlertingPattern](#)
- struct [unpack\\_voice\\_SLQSVoiceGetAllCallInfo\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceManageCalls\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceManageCalls\\_t](#)
- struct [voice\\_burstDTMFInfo](#)
- struct [voice\\_DTMFLengths](#)
- struct [pack\\_voice\\_SLQSVoiceBurstDTMF\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceBurstDTMF\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceStartContDTMF\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceStartContDTMF\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceStopContDTMF\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceStopContDTMF\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceSendFlash\\_t](#)
- struct [unpack\\_voice\\_SLQSVoiceSendFlash\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceSetPreferredPrivacy\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceIndicationRegister\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceGetConfig\\_t](#)
- struct [voice\\_curAMRConfig](#)
- struct [unpack\\_voice\\_SLQSVoiceGetConfig\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceOrigUSSDNoWait\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceBindSubscription\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceALSSetLineSwitching\\_t](#)
- struct [pack\\_voice\\_SLQSVoiceALSSelectLine\\_t](#)
- struct [voice\\_COLPResp](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCOLP\\_t](#)
- struct [voice\\_COLRResp](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCOLR\\_t](#)

- struct [voice\\_CNAPResp](#)
- struct [unpack\\_voice\\_SLQSVoiceGetCNAP\\_t](#)
- struct [pack\\_voice\\_SLQSOriinateUSSD\\_t](#)
- struct [unpack\\_voice\\_SLQSOriinateUSSD\\_t](#)
- struct [voice\\_USSDNotificationNetworkInfo](#)
- struct [unpack\\_voice\\_USSDNotificationCallback\\_ind\\_t](#)
- struct [voice\\_SUPSInfo](#)
- struct [voice\\_newPwdData](#)
- struct [unpack\\_voice\\_SLQSVoiceSUPSCallback\\_ind\\_t](#)
- struct [unpack\\_voice\\_allCallStatusCallback\\_ind\\_t](#)
- struct [unpack\\_voice\\_voicePrivacyChangeCallback\\_ind\\_t](#)
- struct [voice\\_DTMFInfo](#)
- struct [unpack\\_voice\\_DTMFEventCallback\\_ind\\_t](#)
- struct [voice\\_ECTNum](#)
- struct [unpack\\_voice\\_SUPSNotificationCallback\\_ind\\_t](#)
- struct [unpack\\_voice\\_OTASPStatusCallback\\_ind\\_t](#)
- struct [voice\\_signalInfo](#)
- struct [voice\\_callerIDInfo](#)
- struct [voice\\_callingPartyInfo](#)
- struct [voice\\_calledPartyInfo](#)
- struct [voice\\_redirNumInfo](#)
- struct [voice\\_NSSAudioCtrl](#)
- struct [voice\\_lineCtrlInfo](#)
- struct [voice\\_extDispRecInfo](#)
- struct [unpack\\_voice\\_VoiceInfoRecCallback\\_ind\\_t](#)

## Macros

- #define [MAXVOICEUSSDLENGTH](#) 182
- #define [MAX\\_VOICE\\_CALL\\_NO\\_LEN](#) 81
- #define [MAX\\_VOICE\\_DESCRIPTION\\_LENGTH](#) 255
- #define [VOICE\\_MAX\\_NO\\_OF\\_CALLS](#) 20
- #define [BARRING\\_PASSWORD\\_LENGTH](#) 4

## Typedefs

- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_OriginateUSSD\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_AnswerUSSD\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_CancelUSSD\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceSetPreferredPrivacy\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceIndicationRegister\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceOrigUSSDNoWait\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceBindSubscription\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceALSSetLineSwitching\\_t](#)
- typedef [unpack\\_result\\_t](#) [unpack\\_voice\\_SLQSVoiceALSSelectLine\\_t](#)

## Enumerations

- enum `liteServiceClassInformation` {  
`LITE_VOICE_SUPS_SRV_CLASS_NONE` = 0x00,  
`LITE_VOICE_SUPS_SRV_CLASS_VOICE` = 0x01,  
`LITE_VOICE_SUPS_SRV_CLASS_DATA` = 0x02,  
`LITE_VOICE_SUPS_SRV_CLASS_FAX` = 0x04,  
`LITE_VOICE_SUPS_SRV_CLASS_SMS` = 0x08,  
`LITE_VOICE_SUPS_SRV_CLASS_DATA_CIRCUITSYNC` = 0x10,  
`LITE_VOICE_SUPS_SRV_CLASS_DATA_CIRCUITASYNC` = 0x20,  
`LITE_VOICE_SUPS_SRV_CLASS_PACKETACCESS` = 0x40,  
`LITE_VOICE_SUPS_SRV_CLASS_PADACCESS` = 0x80 }

## Functions

- int `pack_voice_OriginateUSSD` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_OriginateUSSD_t` \*reqArg)
- int `unpack_voice_OriginateUSSD` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_OriginateUSSD_t` \*pOutput)
- int `pack_voice_AnswerUSSD` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_AnswerUSSD_t` \*reqArg)
- int `unpack_voice_AnswerUSSD` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_AnswerUSSD_t` \*pOutput)
- int `pack_voice_CancelUSSD` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int `unpack_voice_CancelUSSD` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_CancelUSSD_t` \*pOutput)
- int `pack_voice_SLQSVoiceDialCall` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceDialCall_t` \*pReq)
- int `unpack_voice_SLQSVoiceDialCall` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceDialCall_t` \*pOutput)
- int `pack_voice_SLQSVoiceEndCall` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceEndCall_t` \*pReq)
- int `unpack_voice_SLQSVoiceEndCall` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceEndCall_t` \*pOutput)
- int `pack_voice_SLQSVoiceSetSUPSService` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceSetSUPSService_t` \*pReq)
- int `unpack_voice_SLQSVoiceSetSUPSService` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceSetSUPSService_t` \*pOutput)
- int `pack_voice_SLQSVoiceSetConfig` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceSetConfig_t` \*pReq)
- int `unpack_voice_SLQSVoiceSetConfig` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceSetConfig_t` \*pOutput)
- int `pack_voice_SLQSVoiceAnswerCall` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceAnswerCall_t` \*pReq)
- int `unpack_voice_SLQSVoiceAnswerCall` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceAnswerCall_t` \*pOutput)
- int `pack_voice_SLQSVoiceGetCLIR` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int `unpack_voice_SLQSVoiceGetCLIR` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceGetCLIR_t` \*pOutput)
- int `pack_voice_SLQSVoiceGetCLIP` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int `unpack_voice_SLQSVoiceGetCLIP` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceGetCLIP_t` \*pOutput)
- int `pack_voice_SLQSVoiceGetCallWaiting` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceGetCallWaiting_t` \*pReq)
- int `unpack_voice_SLQSVoiceGetCallWaiting` (uint8\_t \*pResp, uint16\_t respLen, `unpack_voice_SLQSVoiceGetCallWaiting_t` \*pOutput)
- int `pack_voice_SLQSVoiceGetCallBarring` (`pack_qmi_t` \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, `pack_voice_SLQSVoiceGetCallBarring_t` \*pReq)

- [int unpack\\_voice\\_SLQSVoiceGetCallBarring](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCallBarring\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceGetCallForwardingStatus](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceGetCallForwardingStatus\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceGetCallForwardingStatus](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCallForwardingStatus\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceSetCallBarringPassword](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceSetCallBarringPassword\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceSetCallBarringPassword](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceSetCallBarringPassword\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceGetCallInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceGetCallInfo\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceGetCallInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCallInfo\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceGetAllCallInfo](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- [int unpack\\_voice\\_SLQSVoiceGetAllCallInfo](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetAllCallInfo\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceManageCalls](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceManageCalls\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceManageCalls](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceManageCalls\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceBurstDTMF](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceBurstDTMF\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceBurstDTMF](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceBurstDTMF\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceStartContDTMF](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceStartContDTMF\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceStartContDTMF](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceStartContDTMF\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceStopContDTMF](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceStopContDTMF\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceStopContDTMF](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceStopContDTMF\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceSendFlash](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceSendFlash\\_t](#) \*pReq)
- [int unpack\\_voice\\_SLQSVoiceSendFlash](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceSendFlash\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceSetPreferredPrivacy](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceSetPreferredPrivacy\\_t](#) \*reqArg)
- [int unpack\\_voice\\_SLQSVoiceSetPreferredPrivacy](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceSetPreferredPrivacy\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceIndicationRegister](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceIndicationRegister\\_t](#) \*reqArg)
- [int unpack\\_voice\\_SLQSVoiceIndicationRegister](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceIndicationRegister\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceGetConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceGetConfig\\_t](#) \*reqArg)
- [int unpack\\_voice\\_SLQSVoiceGetConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetConfig\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceOrigUSSDNoWait](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceOrigUSSDNoWait\\_t](#) \*reqArg)
- [int unpack\\_voice\\_SLQSVoiceOrigUSSDNoWait](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceOrigUSSDNoWait\\_t](#) \*pOutput)
- [int pack\\_voice\\_SLQSVoiceBindSubscription](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceBindSubscription\\_t](#) \*reqArg)
- [int unpack\\_voice\\_SLQSVoiceBindSubscription](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceBindSubscription\\_t](#) \*pOutput)

- int [pack\\_voice\\_SLQSVoiceALSSetLineSwitching](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceALSSetLineSwitching\\_t](#) \*reqArg)
- int [unpack\\_voice\\_SLQSVoiceALSSetLineSwitching](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceALSSetLineSwitching\\_t](#) \*pOutput)
- int [pack\\_voice\\_SLQSVoiceALSSelectLine](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSVoiceALSSelectLine\\_t](#) \*reqArg)
- int [unpack\\_voice\\_SLQSVoiceALSSelectLine](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceALSSelectLine\\_t](#) \*pOutput)
- int [pack\\_voice\\_SLQSVoiceGetCOLP](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_voice\\_SLQSVoiceGetCOLP](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCOLP\\_t](#) \*pOutput)
- int [pack\\_voice\\_SLQSVoiceGetCOLR](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_voice\\_SLQSVoiceGetCOLR](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCOLR\\_t](#) \*pOutput)
- int [pack\\_voice\\_SLQSVoiceGetCNAP](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_voice\\_SLQSVoiceGetCNAP](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceGetCNAP\\_t](#) \*pOutput)
- int [pack\\_voice\\_SLQSOOriginateUSSD](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_voice\\_SLQSOOriginateUSSD\\_t](#) \*reqArg)
- int [unpack\\_voice\\_SLQSOOriginateUSSD](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSOOriginateUSSD\\_t](#) \*pOutput)
- int [unpack\\_voice\\_USSDNotificationCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_USSDNotificationCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_SLQSVoiceSUPSCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SLQSVoiceSUPSCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_allCallStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_allCallStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_voicePrivacyChangeCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_voicePrivacyChangeCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_DTMFEventCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_DTMFEventCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_SUPSNotificationCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_SUPSNotificationCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_OTASPStatusCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_OTASPStatusCallback\\_ind\\_t](#) \*pOutput)
- int [unpack\\_voice\\_VoiceInfoRecCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_voice\\_VoiceInfoRecCallback\\_ind\\_t](#) \*pOutput)

### 9.44.1 Detailed Description

#### 9.44.2 S1

- 3GPP Mobile Radio Interface Layer 3 Specification : Core Network Protocols; Stage 3 (Release 7)
- 3GPP TS 24.008 V7.0.0 (2005-06)

### 9.44.3 Macro Definition Documentation

9.44.3.1 `#define BARRING_PASSWORD_LENGTH 4`

9.44.3.2 `#define MAX_VOICE_CALL_NO_LEN 81`

9.44.3.3 `#define MAX_VOICE_DESCRIPTION_LENGTH 255`

9.44.3.4 `#define MAXVOICEUSSDLENGTH 182`

9.44.3.5 `#define VOICE_MAX_NO_OF_CALLS 20`

#### 9.44.4 Typedef Documentation

9.44.4.1 `typedef unpack_result_t unpack_voice_AnswerUSSD_t`

9.44.4.2 `typedef unpack_result_t unpack_voice_CancelUSSD_t`

9.44.4.3 `typedef unpack_result_t unpack_voice_OriginateUSSD_t`

9.44.4.4 `typedef unpack_result_t unpack_voice_SLQSVoiceALSSelectLine_t`

9.44.4.5 `typedef unpack_result_t unpack_voice_SLQSVoiceALSSetLineSwitching_t`

9.44.4.6 `typedef unpack_result_t unpack_voice_SLQSVoiceBindSubscription_t`

9.44.4.7 `typedef unpack_result_t unpack_voice_SLQSVoiceIndicationRegister_t`

9.44.4.8 `typedef unpack_result_t unpack_voice_SLQSVoiceOrigUSSDNoWait_t`

9.44.4.9 `typedef unpack_result_t unpack_voice_SLQSVoiceSetPreferredPrivacy_t`

#### 9.44.5 Enumeration Type Documentation

9.44.5.1 `enum liteServiceClassInformation`

Service Class information

Enumerator

***LITE\_VOICE\_SUPS\_SRV\_CLASS\_NONE***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_VOICE***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATA***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_FAX***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_SMS***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITSYNC***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUITASYNC***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS***  
***LITE\_VOICE\_SUPS\_SRV\_CLASS\_PADACCESS***

#### 9.44.6 Function Documentation

9.44.6.1 `int pack_voice_AnswerUSSD ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_AnswerUSSD_t * reqArg )`

Responds to a USSD request from the network 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

#### 9.44.6.2 int pack\_voice\_CancelUSSD ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

Cancels an in-progress USSD operation 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

#### 9.44.6.3 int pack\_voice\_OriginateUSSD ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_OriginateUSSD\_t \* reqArg )

Initiates a USSD operation 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

#### 9.44.6.4 int pack\_voice\_SLQSOriinateUSSD ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSOriinateUSSD\_t \* reqArg )

Initiates a USSD 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>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

9.44.6.5 int pack\_voice\_SLQSVoiceALSSelectLine ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceALSSelectLine\_t \* *reqArg* )

allows the user to select the preferred line 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

9.44.6.6 int pack\_voice\_SLQSVoiceALSSetLineSwitching ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceALSSetLineSwitching\_t \* *reqArg* )

sets the line switch setting on the card 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

9.44.6.7 `int pack_voice_SLQSVoiceAnswerCall ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceAnswerCall_t * pReq )`

Answers an incoming voice call 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>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

9.44.6.8 `int pack_voice_SLQSVoiceBindSubscription ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceBindSubscription_t * reqArg )`

binds a subscription type to a specific voice client 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

9.44.6.9 `int pack_voice_SLQSVoiceBurstDTMF ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceBurstDTMF_t * pReq )`

Sends a burst Dual-Tone Multi frequency (DTMF) 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>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

9.44.6.10 int pack\_voice\_SLQSVoiceDialCall ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSVoiceDialCall\_t \* pReq )

Originates a voice call (MO call) pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReq	request parameter

**Returns**

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

**See Also**

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

9.44.6.11 int pack\_voice\_SLQSVoiceEndCall ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSVoiceEndCall\_t \* pReq )

Ends a voice call pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReq	request parameter

**Returns**

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

**See Also**

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

9.44.6.12 int pack\_voice\_SLQSVoiceGetAllCallInfo ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen )

information associated with all the calls originating or terminating 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

9.44.6.13 int pack\_voice\_SLQSVoiceGetCallBarring ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceGetCallBarring\_t \* *pReq* )

Status of Call Barring Supplementary Service 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>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

9.44.6.14 int pack\_voice\_SLQSVoiceGetCallForwardingStatus ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t \* *pReq* )

Status of Call Forwarding Supplementary Service 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>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

9.44.6.15 `int pack_voice_SLQSVoiceGetCallInfo ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallInfo_t * pReq )`

information associated with a call 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>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

9.44.6.16 `int pack_voice_SLQSVoiceGetCallWaiting ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetCallWaiting_t * pReq )`

Status of Call Waiting Supplementary Service 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>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

9.44.6.17 `int pack_voice_SLQSVoiceGetCLIP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

status of the Calling Line Identification Presentation (CLIP) 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

9.44.6.18 `int pack_voice_SLQSVoiceGetCLIR ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

status of the Calling Line Identification Restriction (CLIR) 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

9.44.6.19 `int pack_voice_SLQSVoiceGetCNAP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

status of the Calling Name Presentation(CNAP) 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

9.44.6.20 `int pack_voice_SLQSVoiceGetCOLP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

status of the Connected Line Identification Presentation 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

9.44.6.21 `int pack_voice_SLQSVoiceGetCOLR ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

status of the Connected Line Identification Restriction 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

9.44.6.22 `int pack_voice_SLQSVoiceGetConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceGetConfig_t * reqArg )`

retrieves various configuration parameters that control the modem behavior 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

9.44.6.23 `int pack_voice_SLQSVoiceIndicationRegister ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceIndicationRegister_t * reqArg )`

Sets the registration state for different QMI\_VOICE indications 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

**9.44.6.24** int pack\_voice\_SLQSVoiceManageCalls ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSVoiceManageCalls\_t \* pReq )

Manages the calls by using the supplementary service pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	pReq	request parameter

**Returns**

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

**See Also**

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

**9.44.6.25** int pack\_voice\_SLQSVoiceOrigUSSDNoWait ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSVoiceOrigUSSDNoWait\_t \* reqArg )

initiates a USSD operation No Wait pack

**Parameters**

in, out	pCtx	qmi request context
out	pReqBuf	qmi request buffer
out	pLen	qmi request length
in	reqArg	request parameter

**Returns**

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

**See Also**

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

**9.44.6.26** int pack\_voice\_SLQSVoiceSendFlash ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_voice\_SLQSVoiceSendFlash\_t \* pReq )

sends a simple flash message 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>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

9.44.6.27 int pack\_voice\_SLQSVoiceSetCallBarringPassword ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceSetCallBarringPassword\_t \* *pReq* )

Sets a Call Barring Password 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>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

9.44.6.28 int pack\_voice\_SLQSVoiceSetConfig ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*, pack\_voice\_SLQSVoiceSetConfig\_t \* *pReq* )

configuration parameters that control the modem behavior 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>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

9.44.6.29 `int pack_voice_SLQSVoiceSetPreferredPrivacy ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceSetPreferredPrivacy_t * reqArg )`

sets the voice privacy preference 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

9.44.6.30 `int pack_voice_SLQSVoiceSetSUPSService ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceSetSUPSService_t * pReq )`

call-independent supplementary services 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>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

9.44.6.31 `int pack_voice_SLQSVoiceStartContDTMF ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_voice_SLQSVoiceStartContDTMF_t * pReq )`

Starts a continuous DTMF 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>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

**9.44.6.32** int pack\_voice\_SLQSVoiceStopContDTMF ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_voice\_SLQSVoiceStopContDTMF\_t \* *pReq* )

Stops a continuous DTMF 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>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

**9.44.6.33** int unpack\_voice\_allCallStatusCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_allCallStatusCallback\_ind\_t \* *pOutput* )

Unpack all call status callback indication.

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	all call status callback unpacked

**Returns**

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

**See Also**

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

**9.44.6.34** int unpack\_voice\_AnswerUSSD ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_AnswerUSSD\_t \* *pOutput*  
)

Responds to a USSD request from the 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

9.44.6.35 int unpack\_voice\_CancelUSSD ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_CancelUSSD\_t \* *pOutput* )

Cancels an in-progress USSD operation 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

9.44.6.36 int unpack\_voice\_DTMFEventCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_DTMFEventCallback\_ind\_t \* *pOutput* )

Unpack DTMF event callback indication.

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	DTMF event callback unpacked

## Returns

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

## See Also

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

9.44.6.37 int unpack\_voice\_OriginateUSSD ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_OriginateUSSD\_t \* *pOutput* )

Initiates a USSD operation 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

**9.44.6.38** int unpack\_voice\_OTASPStatusCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_OTASPStatusCallback\_ind\_t \* *pOutput* )

Unpack voice OTASP status callback indication.

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	voice OTASP status callback unpacked

**Returns**

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

**See Also**

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

**9.44.6.39** int unpack\_voice\_SLQSOriinateUSSD ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSOriinateUS-  
SD\_t \* *pOutput* )

Initiates a USSD 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

9.44.6.40 `int unpack_voice_SLQSVoiceALSSelectLine ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceALSSelectLine_t * pOutput )`

allows the user to select the preferred line 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

9.44.6.41 `int unpack_voice_SLQSVoiceALSSetLineSwitching ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceALSSetLineSwitching_t * pOutput )`

sets the line switch setting on the card 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

9.44.6.42 `int unpack_voice_SLQSVoiceAnswerCall ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceAnswerCall_t * pOutput )`

Answers an incoming voice call 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

9.44.6.43 `int unpack_voice_SLQSVoiceBindSubscription ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceBindSubscription_t * pOutput )`

binds a subscription type to a specific voice client ID 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

9.44.6.44 `int unpack_voice_SLQSVoiceBurstDTMF ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceBurstDTMF_t * pOutput )`

Sends a burst Dual-Tone Multi frequency (DTMF) 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

9.44.6.45 `int unpack_voice_SLQSVoiceDialCall ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceDialCall_t * pOutput )`

Originates a voice call (MO call) 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

9.44.6.46 int unpack\_voice\_SLQSVoiceEndCall ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceEndCall\_t \* *pOutput* )

Ends a voice call 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

9.44.6.47 int unpack\_voice\_SLQSVoiceGetAllCallInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceGetAllCallInfo\_t \* *pOutput* )

information associated with all the calls originating or terminating 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

9.44.6.48 int unpack\_voice\_SLQSVoiceGetCallBarring ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceGetCallBarring\_t \* *pOutput* )

Status of Call Barring Supplementary Service 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

**9.44.6.49** int unpack\_voice\_SLQSVoiceGetCallForwardingStatus ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t \* *pOutput* )

Status of Call Forwarding Supplementary Service 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

**9.44.6.50** int unpack\_voice\_SLQSVoiceGetCallInfo ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceGet-  
CallInfo\_t \* *pOutput* )

information associated with a call 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

**9.44.6.51** int unpack\_voice\_SLQSVoiceGetCallWaiting ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceGetCallWaiting\_t \* *pOutput* )

Status of Call Waiting Supplementary Service 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

**9.44.6.52** `int unpack_voice_SLQSVoiceGetCLIP ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIP_t * pOutput )`

status of the Calling Line Identification Presentation (CLIP) 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

**9.44.6.53** `int unpack_voice_SLQSVoiceGetCLIR ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCLIR_t * pOutput )`

status of the Calling Line Identification Restriction (CLIR) 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

**9.44.6.54** `int unpack_voice_SLQSVoiceGetCNAP ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCNAP_t * pOutput )`

status of the Calling Name Presentation (CNAP) 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

9.44.6.55 `int unpack_voice_SLQSVoiceGetCOLP ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLP_t * pOutput )`

status of the Connected Line Identification Presentation 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

9.44.6.56 `int unpack_voice_SLQSVoiceGetCOLR ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetCOLR_t * pOutput )`

status of the Connected Line Identification Restriction 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

9.44.6.57 `int unpack_voice_SLQSVoiceGetConfig ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceGetConfig_t * pOutput )`

retrieves various configuration parameters that control the modem behavior 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

**9.44.6.58** int unpack\_voice\_SLQSVoiceIndicationRegister ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceIndicationRegister\_t \* *pOutput* )

Sets the registration state for different QMI\_VOICE indications 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

**9.44.6.59** int unpack\_voice\_SLQSVoiceManageCalls ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceManageCalls\_t \* *pOutput* )

Manages the calls by using the supplementary service 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

**9.44.6.60** int unpack\_voice\_SLQSVoiceOrigUSSDNoWait ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceOrigUSSDNoWait\_t \* *pOutput* )

initiates a USSD operation No Wait 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

**9.44.6.61** int unpack\_voice\_SLQSVoiceSendFlash ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceSendFlash\_t \* *pOutput* )

sends a simple flash message 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

**9.44.6.62** int unpack\_voice\_SLQSVoiceSetCallBarringPassword ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceSetCallBarringPassword\_t \* *pOutput* )

Sets a Call Barring Password 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

**9.44.6.63** int unpack\_voice\_SLQSVoiceSetConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_voice\_SLQSVoiceSetConfig\_t \* *pOutput* )

configuration parameters that control the modem behavior 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

9.44.6.64 `int unpack_voice_SLQSVoiceSetPreferredPrivacy ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetPreferredPrivacy_t * pOutput )`

sets the voice privacy 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

9.44.6.65 `int unpack_voice_SLQSVoiceSetSUPSService ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceSetSUPSService_t * pOutput )`

call-independent supplementary services 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

9.44.6.66 `int unpack_voice_SLQSVoiceStartContDTMF ( uint8_t * pResp, uint16_t respLen, unpack_voice_SLQSVoiceStartContDTMF_t * pOutput )`

Starts a continuous DTMF 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

9.44.6.67 int unpack\_voice\_SLQSVoiceStopContDTMF ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceStopContDTMF\_t \* *pOutput* )

Stops a continuous DTMF 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

**Note**

return eQCWWAN\_ERR\_NULL\_TLV then callID value is invalid.

**See Also**

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

9.44.6.68 int unpack\_voice\_SLQSVoiceSUPSCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t \* *pOutput* )

Unpack voice sups callback indication.

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	voice susp callback unpacked

**Returns**

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

**See Also**

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

9.44.6.69 int unpack\_voice\_SUPSNotificationCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_SUPSNotificationCallback\_ind\_t \* *pOutput* )

Unpack SUPS notification callback indication.



## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	SUPS notification callback unpacked

## Returns

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

## See Also

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

9.44.6.70 int unpack\_voice\_USSDNotificationCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_USSDNotificationCallback\_ind\_t \* *pOutput* )

Unpack USSD notification callback indication.

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	USSD notification unpacked

## Returns

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

## See Also

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

9.44.6.71 int unpack\_voice\_VoiceInfoRecCallback\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_voice\_VoiceInfoRecCallback\_ind\_t \* *pOutput* )

Unpack SUPS notification callback indication.

## Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	SUPS notification callback unpacked

## Returns

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

## See Also

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

9.44.6.72 `int unpack_voice_voicePrivacyChangeCallback_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_voice_voicePrivacyChangeCallback_ind_t * pOutput )`

Unpack voice privacy change callback indication.

#### Parameters

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	voice privacy change callback unpacked

#### Returns

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

#### See Also

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

## 9.45 wds.h File Reference

### Data Structures

- struct [LibPackQosClassID](#)
- struct [LibPackTFTIDParams](#)
- struct [LibPackGPRSRequestedQoS](#)
- struct [LibPackUMTSQoS](#)
- struct [LibPackUMTSReqQoSSigInd](#)
- struct [LibPackProfileMnc](#)
- struct [LibPackPDNThrottleTimer](#)
- struct [LibPackPCOIDList](#)
- struct [pack\\_wds\\_SLQSStartDataSession\\_t](#)
- struct [unpack\\_wds\\_SLQSStartDataSession\\_t](#)
- struct [unpack\\_wds\\_SLQSSetPacketSrvStatusCallback\\_t](#)
- struct [pack\\_wds\\_SLQSStopDataSession\\_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 [LibpackProfile3GPPV2](#)
- union [unpackWdsProfileParamV2](#)
- struct [UnPackGetProfileSettingOutV2](#)
- struct [unpack\\_wds\\_SLQSGetProfileSettingsV2\\_t](#)
- struct [unpack\\_wds\\_GetSessionState\\_t](#)
- struct [pack\\_wds\\_GetDefaultProfile\\_t](#)
- struct [unpack\\_wds\\_GetDefaultProfile\\_t](#)
- struct [pack\\_wds\\_GetDefaultProfileV2\\_t](#)
- struct [unpack\\_wds\\_GetDefaultProfileV2\\_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\\_GetSessionDurationV2\\_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 [wds\\_transferStatInd](#)
- struct [pack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [connectionStatus](#)
- struct [dunchannelRate](#)
- struct [unpack\\_wds\\_SLQSGetDUNCallInfo\\_t](#)
- struct [qmiWSDDataBearerTechnology](#)

- 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](#)
- struct [pack\\_wds\\_GetPacketStatistics\\_t](#)
- struct [unpack\\_wds\\_GetPacketStatistics\\_t](#)
- struct [unpack\\_wds\\_GetByteTotals\\_t](#)
- struct [unpack\\_wds\\_SLQSGetCurrentChannelRate\\_t](#)
- struct [unpack\\_wds\\_SLQSSetLoopback\\_t](#)
- struct [pack\\_wds\\_SLQSSetLoopback\\_t](#)
- struct [wds\\_DataULongTlv](#)
- struct [wds\\_DataULongLongTlv](#)
- struct [unpack\\_RMTransferStatistics\\_ind\\_t](#)
- struct [pack\\_wds\\_DHCPv4ClientLeaseChange\\_t](#)
- struct [wds\\_DHCPProfileIdTlv](#)
- struct [wds\\_DHCPLeaseStateTlv](#)
- struct [wds\\_IPv4AdTlv](#)
- struct [wds\\_DHCPOpt](#)
- struct [wds\\_DHCPLeaseOptTlv](#)
- struct [unpack\\_wds\\_DHCPv4ClientLease\\_ind\\_t](#)
- struct [pack\\_wds\\_SetMobileIP\\_t](#)
- struct [pack\\_wds\\_SetMobileIPParameters\\_t](#)
- struct [pack\\_wds\\_SetAutoconnect\\_t](#)
- struct [unpack\\_wds\\_GetAutoconnect\\_t](#)
- struct [wds\\_TrStatInd](#)
- struct [pack\\_wds\\_SLQSWdsSetEventReport\\_t](#)
- struct [wds\\_DHCPv4ProfileId](#)
- struct [wds\\_DHCPv4HWConfig](#)
- struct [wds\\_DHCPv4Option](#)
- struct [wds\\_DHCPv4OptionList](#)
- struct [pack\\_wds\\_SLQSSetDHCPv4ClientConfig\\_t](#)
- struct [unpack\\_wds\\_GetDataBearerTechnology\\_t](#)
- struct [wds\\_ConnStatusTlv](#)
- struct [wds\\_LastMdmCallEndRsnTlv](#)
- struct [wds\\_TXBytesOKTlv](#)
- struct [wds\\_RXBytesOKTlv](#)
- struct [wds\\_DormStatTlv](#)
- struct [wds\\_DataBearTechTlv](#)
- struct [wds\\_channelRate](#)
- struct [wds\\_ChannelRateTlv](#)
- struct [unpack\\_wds\\_SLQSDUNCallInfoCallBack\\_ind\\_t](#)
- struct [pack\\_wds\\_SLQSSwiProfileChangeCallback\\_t](#)
- struct [wds\\_profileChange](#)
- struct [wds\\_sourceOfChange](#)
- struct [unpack\\_wds\\_SLQSSwiProfileChangeCallback\\_Ind\\_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`
- `#define BYT_STAT_STAT_MASK 0X000000C0`
- `#define WDS_DHCP_MAX_NUM_OPTIONS 30`
- `#define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */`
- `#define WDS_TFTID_SOURCE_IP_SIZE 8`
- `#define WDS_PROFILE_3GPP 0`
- `#define WDS_PROFILE_3GPP2 1`
- `#define LITE_MAX_PDN_THROTTLE_TIMER 10`
- `#define LITE_MAX_PCID_LIST 10`
- `#define MAX_WDS_NAME_ARRAY_SIZE 255`

## Typedefs

- `typedef unpack_result_t unpack_wds_SLQSSStopDataSession_t`
- `typedef unpack_result_t unpack_wds_SLQSSetWdsEventCallback_t`
- `typedef union unpackWdsProfileParam UnpackQmiProfileInfo`
- `typedef union unpackWdsProfileParamV2 UnpackQmiProfileInfoV2`
- `typedef unpack_result_t unpack_wds_SetDefaultProfile_t`
- `typedef unpack_result_t unpack_wds_SLQSSet3GPPConfigItem_t`
- `typedef unpack_result_t unpack_wds_SetDefaultProfileNum_t`
- `typedef unpack_result_t unpack_wds_SLQSSetLoopback_t`
- `typedef struct unpack_RMTransferStatistics_ind_t unpack_wds_RMTransferStatistics_ind_t`
- `typedef unpack_result_t unpack_wds_DHCPv4ClientLeaseChange_t`
- `typedef unpack_result_t unpack_wds_SetMobileIP_t`
- `typedef unpack_result_t unpack_wds_SetMobileIPParameters_t`
- `typedef unpack_result_t unpack_wds_SetAutoconnect_t`
- `typedef unpack_result_t unpack_wds_SLQSWdsSetEventReport_t`
- `typedef unpack_result_t unpack_wds_SLQSWdsGoDormant_t`
- `typedef unpack_result_t unpack_wds_SLQSWdsGoActive_t`
- `typedef unpack_result_t unpack_wds_SLQSResetPacketStatics_t`
- `typedef unpack_result_t unpack_wds_SLQSSetDHCPv4ClientConfig_t`
- `typedef unpack_result_t unpack_wds_SetMuxID_t`
- `typedef unpack_result_t unpack_wds_SLQSSwiProfileChangeCallback_t`

## Enumerations

- `enum liteQmiDataBearerMasks {`  
`QMI_LITE_WDS_CURRENT_CALL_DB_MASK = 0x01,`  
`QMI_LITE_WDS_LAST_CALL_DB_MASK = 0x02 }`

## 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, unpack_wds_SLQSStopDataSession_t *pOutput)`
- `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, unpack_wds_SLQSSetWdsEventCallback_t *pOutput)`
- `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_SLQSGetProfileSettingsV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSGetProfileSettingsV2_t *reqArg)`
- `int unpack_wds_SLQSGetProfileSettingsV2 (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetProfileSettingsV2_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_GetDefaultProfileV2 (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetDefaultProfileV2_t *reqParam)`
- `int unpack_wds_GetDefaultProfileV2 (uint8_t *pResp, uint16_t respLen, unpack_wds_GetDefaultProfileV2_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\\_GetSessionDurationV2](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_GetSessionDuration\\_t](#) \*reqParam)
- int [unpack\\_wds\\_GetSessionDurationV2](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetSessionDurationV2\\_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, [unpack\\_wds\\_SetDefaultProfile\\_t](#) \*pOutput)
- 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, [unpack\\_wds\\_SLQSSet3GPPConfigItem\\_t](#) \*pOutput)
- 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\\_SLQSWdsSwiPDPRuntimeSettings](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_t](#) \*reqParam)
- int [unpack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSWdsSwiPDPRuntimeSettings\\_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, unpack_wds_SetDefaultProfileNum_t *pOutput)`
- `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)`
- `int pack_wds_GetPacketStatistics (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_GetPacketStatistics_t *pReq)`
- `int unpack_wds_GetPacketStatistics (uint8_t *pResp, uint16_t respLen, unpack_wds_GetPacketStatistics_t *pOutput)`
- `int pack_wds_GetByteTotals (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_wds_GetByteTotals (uint8_t *pResp, uint16_t respLen, unpack_wds_GetByteTotals_t *pOutput)`
- `int pack_wds_SLQSGetCurrentChannelRate (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_wds_SLQSGetCurrentChannelRate (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSGetCurrentChannelRate_t *pOutput)`
- `int pack_wds_SLQSSetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen)`
- `int unpack_wds_SLQSSetLoopback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetLoopback_t *pOutput)`
- `int pack_wds_SLQSSetLoopback (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SLQSSetLoopback_t *reqArg)`
- `int unpack_wds_SLQSSetLoopback (uint8_t *pResp, uint16_t respLen, unpack_wds_SLQSSetLoopback_t *pOutput)`
- `int unpack_wds_RMTransferStatistics_ind (uint8_t *pResp, uint16_t respLen, unpack_RMTransferStatistics_ind_t *pOutput)`
- `int pack_wds_DHCPv4ClientLeaseChange (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_DHCPv4ClientLeaseChange_t *reqArg)`
- `int unpack_wds_DHCPv4ClientLeaseChange (uint8_t *pResp, uint16_t respLen, unpack_wds_DHCPv4ClientLeaseChange_t *pOutput)`
- `int unpack_wds_DHCPv4ClientLease_ind (uint8_t *pResp, uint16_t respLen, unpack_wds_DHCPv4ClientLease_ind_t *pOutput)`
- `int pack_wds_SetMobileIP (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIP_t *reqArg)`
- `int unpack_wds_SetMobileIP (uint8_t *pResp, uint16_t respLen, unpack_wds_SetMobileIP_t *pOutput)`
- `int pack_wds_SetMobileIPParameters (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetMobileIPParameters_t *reqArg)`
- `int unpack_wds_SetMobileIPParameters (uint8_t *pResp, uint16_t respLen, unpack_wds_SetMobileIPParameters_t *pOutput)`
- `int pack_wds_SetAutoconnect (pack_qmi_t *pCtx, uint8_t *pReqBuf, uint16_t *pLen, pack_wds_SetAutoconnect_t *reqArg)`



- int [unpack\\_wds\\_SetAutoconnect](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SetAutoconnect\\_t](#) \*pOutput)
- int [pack\\_wds\\_GetAutoconnect](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_GetAutoconnect](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetAutoconnect\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSWdsSetEventReport](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSWdsSetEventReport\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSWdsSetEventReport](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSWdsSetEventReport\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSWdsGoDormant](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSWdsGoDormant](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSWdsGoDormant\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSWdsGoActive](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSWdsGoActive](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSWdsGoActive\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSRResetPacketStatics](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_SLQSRResetPacketStatics](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSRResetPacketStatics\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSSSetDHCPv4ClientConfig](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSSetDHCPv4ClientConfig\\_t](#) \*pOutput)
- int [pack\\_wds\\_GetDataBearerTechnology](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen)
- int [unpack\\_wds\\_GetDataBearerTechnology](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_GetDataBearerTechnology\\_t](#) \*pOutput)
- int [pack\\_wds\\_SetMuxID](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, uint8\_t \*pMuxID)
- int [unpack\\_wds\\_SetMuxID](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SetMuxID\\_t](#) \*pOutput)
- int [unpack\\_wds\\_SLQSDUNCallInfoCallback\\_ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSDUNCallInfoCallback\\_ind\\_t](#) \*pOutput)
- int [pack\\_wds\\_SLQSSwiProfileChangeCallback](#) ([pack\\_qmi\\_t](#) \*pCtx, uint8\_t \*pReqBuf, uint16\_t \*pLen, [pack\\_wds\\_SLQSSwiProfileChangeCallback\\_t](#) \*reqArg)
- int [unpack\\_wds\\_SLQSSwiProfileChangeCallback](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSwiProfileChangeCallback\\_t](#) \*pOutput)
- int [unpack\\_wds\\_SLQSSwiProfileChangeCallback\\_Ind](#) (uint8\_t \*pResp, uint16\_t respLen, [unpack\\_wds\\_SLQSSwiProfileChangeCallback\\_Ind\\_t](#) \*pOutput)

### 9.45.1 Detailed Description

### 9.45.2 SO Mask

- SO mask to indicate the service option or type of application.
  - An SO mask value of zero indicates that this field is ignored.
  - Values:
    - \* 0x00 - DONT\_CARE
  - CDMA 1X SO mask:
    - \* 0x01 - CDMA\_1X\_IS95
    - \* 0x02 - CDMA\_1X\_IS2000
    - \* 0x04 - CDMA\_1X\_IS2000\_REL\_A
  - CDMA EV-DO Rev 0 SO mask:
    - \* 0x01 - DPA
  - CDMA EV-DO Rev A SO mask:
    - \* 0x01 - DPA

- \* 0x02 - MFPA
- \* 0x04 - EMPA
- \* 0x08 - EMPA\_EHRPD
- CDMA EV-DO Rev B SO mask:
  - \* 0x01 - DPA
  - \* 0x02 - MFPA
  - \* 0x04 - EMPA
  - \* 0x08 - EMPA\_EHRPD
  - \* 0x10 - MMPA
  - \* 0x20 - MMPA\_EHRPD

### 9.45.3 RAT Mask

- RAT mask to indicate the type of technology. A RAT mask value of zero indicates that this field is ignored.

Values:

- 0x00 - DONT\_CARE
- 0x8000 - NULL\_BEARER
- CDMA RAT mask:
  - 0x01 - CDMA\_1X
  - 0x02 - EVDO\_REV0
  - 0x04 - EVDO\_REVA
  - 0x08 - EVDO\_REVB
  - 0x10 - EHRPD
  - 0x20 - FMC
- UMTS RAT mask:
  - 0x01 - WCDMA
  - 0x02 - GPRS
  - 0x04 - HSDPA
  - 0x08 - HSUPA
  - 0x10 - EDGE
  - 0x20 - LTE
  - 0x40 - HSDPA+
  - 0x80 - DC\_HSDPA+
  - 0x100 - 64\_QAM
  - 0x200 - TD-SCDMA

### 9.45.4 Macro Definition Documentation

9.45.4.1 `#define BYT_STAT_STAT_MASK 0X000000C0`

9.45.4.2 `#define IPV6_ADDRESS_ARRAY_SIZE 8`

9.45.4.3 `#define LITE_MAX_PCOID_LIST 10`

9.45.4.4 `#define LITE_MAX_PDN_THROTTLE_TIMER 10`

9.45.4.5 `#define MAX_WDS_3GPP_CONF_LTE_ATTACH_PROFILE_LIST_SIZE 24`

9.45.4.6 `#define MAX_WDS_NAME_ARRAY_SIZE 255`

9.45.4.7 `#define PACK_WDS_IPV4 4`

9.45.4.8 `#define PACK_WDS_IPV6 6`

9.45.4.9 `#define WDS_DHCP_MAX_NUM_OPTIONS 30`

9.45.4.10 `#define WDS_DHCP_OPTION_DATA_BUF_SIZE 2048 /* current max size of raw message in SDK process is 2048 */`

9.45.4.11 `#define WDS_PROFILE_3GPP 0`

9.45.4.12 `#define WDS_PROFILE_3GPP2 1`

9.45.4.13 `#define WDS_TFTID_SOURCE_IP_SIZE 8`

## 9.45.5 Typedef Documentation

9.45.5.1 `typedef unpack_result_t unpack_wds_DHCPv4ClientLeaseChange_t`

9.45.5.2 `typedef struct unpack_RMTransferStatistics_ind_t unpack_wds_RMTransferStatistics_ind_t`

9.45.5.3 `typedef unpack_result_t unpack_wds_SetAutoconnect_t`

9.45.5.4 `typedef unpack_result_t unpack_wds_SetDefaultProfile_t`

9.45.5.5 `typedef unpack_result_t unpack_wds_SetDefaultProfileNum_t`

9.45.5.6 `typedef unpack_result_t unpack_wds_SetMobileIP_t`

9.45.5.7 `typedef unpack_result_t unpack_wds_SetMobileIPParameters_t`

9.45.5.8 `typedef unpack_result_t unpack_wds_SetMuxID_t`

9.45.5.9 `typedef unpack_result_t unpack_wds_SLQSRResetPacketStatics_t`

9.45.5.10 `typedef unpack_result_t unpack_wds_SLQSSet3GPPConfigItem_t`

9.45.5.11 `typedef unpack_result_t unpack_wds_SLQSSetWdsEventCallback_t`

9.45.5.12 `typedef unpack_result_t unpack_wds_SLQSSetDHCPv4ClientConfig_t`

9.45.5.13 `typedef unpack_result_t unpack_wds_SLQSSetLoopback_t`

9.45.5.14 `typedef unpack_result_t unpack_wds_SLQSSetStopDataSession_t`

9.45.5.15 `typedef unpack_result_t unpack_wds_SLQSSetProfileChangeCallback_t`

9.45.5.16 `typedef unpack_result_t unpack_wds_SLQSWdsGoActive_t`

9.45.5.17 `typedef unpack_result_t unpack_wds_SLQSWdsGoDormant_t`

9.45.5.18 `typedef unpack_result_t unpack_wds_SLQSWdsSetEventReport_t`

#### 9.45.5.19 typedef union unpackWdsProfileParam UnpackQmiProfileInfo

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

##### Parameters

<i>SlqsProfile3GPP</i>	3GPP profile <ul style="list-style-type: none"> <li>See <a href="#">LibpackProfile3GPP</a></li> </ul>
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> <li>See <a href="#">LibpackProfile3GPP2</a></li> </ul>

#### 9.45.5.20 typedef union unpackWdsProfileParamV2 UnpackQmiProfileInfoV2

This union WdsProfileParam consists of Profile3GPP and Profile3GPP2 out of which one will be used to create profile.

##### Parameters

<i>SlqsProfile3GPP</i>	3GPP profile <ul style="list-style-type: none"> <li>See <a href="#">LibpackProfile3GPP</a></li> </ul>
<i>SlqsProfile3GPP2</i>	3GPP2 profile <ul style="list-style-type: none"> <li>See <a href="#">LibpackProfile3GPP2</a></li> </ul>

### 9.45.6 Enumeration Type Documentation

#### 9.45.6.1 enum liteQmiDataBearerMasks

Bit mask values to indicate the presence of data bearer information for the current and last data calls

##### Enumerator

***QMI\_LITE\_WDS\_CURRENT\_CALL\_DB\_MASK***  
***QMI\_LITE\_WDS\_LAST\_CALL\_DB\_MASK***

### 9.45.7 Function Documentation

#### 9.45.7.1 int pack\_wds\_DHCPv4ClientLeaseChange ( pack\_qmi\_t \* pCtx, uint8\_t \* pReqBuf, uint16\_t \* pLen, pack\_wds\_DHCPv4ClientLeaseChange\_t \* reqArg )

DHCPv4 lease state changes 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

**Note**

PDN Specific: Yes

### 9.45.7.2 int pack\_wds\_GetAutoconnect ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets auto connect data session setting 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

### 9.45.7.3 int pack\_wds\_GetByteTotals ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

get Rx/Tx byte counts since the start of the last packet 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

**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

9.45.7.4 `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>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

9.45.7.5 `int pack_wds_GetDataBearerTechnology ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get current data bearer technology 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

9.45.7.6 `int pack_wds_GetDefaultProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfile_t * reqParam )`

get default profile 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

**9.45.7.7** `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request parameter

**Returns**

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

**See Also**

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

**9.45.7.8** `int pack_wds_GetDefaultProfileV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_GetDefaultProfileV2_t * reqParam )`

get default profile pack V2.

**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

9.45.7.9 `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>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: No

9.45.7.10 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</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: No

9.45.7.11 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</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: No

**9.45.7.12** 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>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

**9.45.7.13** int pack\_wds\_GetPacketStatistics ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_GetPacketStatistics\_t \* *pReq* )

gets current packet transfer counter values 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>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

9.45.7.14 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>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

9.45.7.15 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>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: No

9.45.7.16 `int pack_wds_GetSessionDurationV2 ( 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>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: No

9.45.7.17 `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>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

9.45.7.18 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>reqParam</i>	request parameter

## Note

PDN Specific: No

```
9.45.7.19 int pack_wds_SetAutoconnect ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
                                     pack_wds_SetAutoconnect_t * reqArg )
```

Auto connect data session parameters 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

## Note

PDN Specific: Yes

```
9.45.7.20 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>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

9.45.7.21 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request parameter

#### Returns

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

#### See Also

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

9.45.7.22 `int pack_wds_SetMobileIP ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SetMobileIP_t * reqArg )`

Sets the current mobile IP setting 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

#### Note

PDN Specific: Yes

9.45.7.23 `int pack_wds_SetMobileIPParameters ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SetMobileIPParameters_t * reqArg )`

Sets the specified mobile IP parameters 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

**Note**

PDN Specific: Yes

9.45.7.24 `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>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

9.45.7.25 `int pack_wds_SetMuxID ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, uint8_t * pMuxID )`

Set MUX 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>pMuxID</i>	MUX ID. • 0x80 to 0x88

**Returns**

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

## See Also

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

9.45.7.26 `int pack_wds_SLQSCreateProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSCreateProfile_t * reqArg )`

Create Profile 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

## Note

PDN Specific: Yes

9.45.7.27 `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>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

9.45.7.28 `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>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

9.45.7.29 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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</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

9.45.7.30 int pack\_wds\_SLQSGetCurrentChannelRate ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

get current Tx/Rx channel bitrate of the current packet data 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

9.45.7.31 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</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

9.45.7.32 `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>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

9.45.7.33 `int pack_wds_SLQSGetProfileSettings ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSGetProfileSettings_t * reqArg )`

Get Profile 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
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

**Note**

PDN Specific: Yes

9.45.7.34 `int pack_wds_SLQSGetProfileSettingsV2 ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSGetProfileSettings_t * reqArg )`

Get Profile Settings pack V2

**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

**Note**

PDN Specific: Yes

9.45.7.35 `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
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

#### Note

PDN Specific: Yes

9.45.7.36 `int pack_wds_SLQSModifyProfile ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSModifyProfile_t * reqArg )`

Modify Profile 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

#### Note

PDN Specific: Yes

9.45.7.37 `int pack_wds_SLQSResetPacketStatics ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

Reset packet data transfer 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

**Note**

PDN Specific: Yes

9.45.7.38 `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>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

9.45.7.39 `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>pReqBuf</i>	qmi request buffer
out	<i>pLen</i>	qmi request length
in	<i>pReqParam</i>	request parameter

**Returns**

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

**See Also**

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

9.45.7.40 `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
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

#### Note

PDN Specific: No

9.45.7.41 `int pack_wds_SLQSSGetDHCPv4ClientConfig ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSGetDHCPv4ClientConfig_t * pReq )`

get DHCPv4 Client 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
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

9.45.7.42 `int pack_wds_SLQSSGetLoopback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen )`

get the value of loopback mode and multiplier 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

**9.45.7.43** int pack\_wds\_SLQSSSetDHCPv4ClientConfig ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t \* *reqArg* )

Gets the DHCP Client V4 Configuration 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

**9.45.7.44** int pack\_wds\_SLQSSSetLoopback ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen*,  
pack\_wds\_SLQSSSetLoopback\_t \* *reqArg* )

Enable/disable Data Loopback Mode and set the value of loopback multiplier 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

**Note**

PDN Specific: Yes

9.45.7.45 `int pack_wds_SLQSSstartDataSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSstartDataSession_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>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

#### Note

PDN Specific: Yes

9.45.7.46 `int pack_wds_SLQSSstopDataSession ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSstopDataSession_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>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

#### Note

PDN Specific: Yes

9.45.7.47 `int pack_wds_SLQSSwiProfileChangeCallback ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen, pack_wds_SLQSSwiProfileChangeCallback_t * reqArg )`

Gets profile change information 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

9.45.7.48 int pack\_wds\_SLQSWdsGoActive ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets the device into Active 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

**Note**

PDN Specific: Yes

9.45.7.49 int pack\_wds\_SLQSWdsGoDormant ( pack\_qmi\_t \* *pCtx*, uint8\_t \* *pReqBuf*, uint16\_t \* *pLen* )

Gets the device into dormant 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



## Note

PDN Specific: Yes

```
9.45.7.50 int pack_wds_SLQSWdsSetEventReport ( pack_qmi_t * pCtx, uint8_t * pReqBuf, uint16_t * pLen,
        pack_wds_SLQSWdsSetEventReport_t * reqArg )
```

Sets the event report parameters 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

## Note

PDN Specific: Yes

```
9.45.7.51 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>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

9.45.7.52 `int unpack_wds_DHCPv4ClientLease_ind ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_DHCPv4ClientLease_ind_t * pOutput )`

DHCP lease state has changed indication 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

9.45.7.53 `int unpack_wds_DHCPv4ClientLeaseChange ( uint8_t * pResp, uint16_t respLen,  
unpack_wds_DHCPv4ClientLeaseChange_t * pOutput )`

DHCPv4 lease state changes unpack

#### 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

9.45.7.54 `int unpack_wds_GetAutoconnect ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetAutoconnect_t *  
pOutput )`

Gets auto connect data session setting 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

**9.45.7.55** `int unpack_wds_GetByteTotals ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetByteTotals_t * pOutput )`

get Rx/Tx byte counts since the start of the last packet 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

**9.45.7.56** `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

**9.45.7.57** `int unpack_wds_GetDataBearerTechnology ( uint8_t * pResp, uint16_t respLen, unpack_wds_GetDataBearerTechnology_t * pOutput )`

get current 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

**9.45.7.58** int unpack\_wds\_GetDefaultProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetDefaultProfile\_t \* *pOutput* )

get 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

**9.45.7.59** 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

**9.45.7.60** int unpack\_wds\_GetDefaultProfileV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetDefaultProfileV2\_t \* *pOutput* )

get default profile unpack V2.

**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

**9.45.7.61** 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

**9.45.7.62** 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

**9.45.7.63** 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

**9.45.7.64** 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

**9.45.7.65** int unpack\_wds\_GetPacketStatistics ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetPacketStatistics\_t \* *pOutput* )

gets current packet transfer counter values 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

**9.45.7.66** 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

9.45.7.67 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

9.45.7.68 int unpack\_wds\_GetSessionDurationV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_GetSessionDurationV2\_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

9.45.7.69 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

**9.45.7.70** 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
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

**9.45.7.71** int unpack\_wds\_RMTransferStatistics\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_RMTransferStatistics\_ind\_t \* *pOutput* )

RM transfer statistics indication 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

**9.45.7.72** int unpack\_wds\_SetAutoconnect ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetAutoconnect\_t \*  
*pOutput* )

Auto connect data session parameters unpack

**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

**9.45.7.73** int unpack\_wds\_SetDefaultProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetDefaultProfile\_t \* *pOutput* )

set default profile unpack

## 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

**9.45.7.74** int unpack\_wds\_SetDefaultProfileNum ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetDefaultProfile-Num\_t \* *pOutput* )

set default profile number unpack

## 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

**9.45.7.75** int unpack\_wds\_SetMobileIP ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetMobileIP\_t \* *pOutput* )

Sets the current mobile IP setting unpack

## 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

**9.45.7.76** int unpack\_wds\_SetMobileIPParameters ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetMobileIPParameters\_t \* *pOutput* )

Sets the specified mobile IP parameters unpack

**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

**9.45.7.77** 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
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

**9.45.7.78** int unpack\_wds\_SetMuxID ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SetMuxID\_t \* *pOutput* )

Set MUX ID unpack.

**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

**9.45.7.79** int unpack\_wds\_SLQSCreateProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSCreateProfile\_t \* *pOutput* )

Create Profile unpack

## 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

**9.45.7.80** 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

**9.45.7.81** int unpack\_wds\_SLQSDUNCallInfoCallBack\_ind ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t \* *pOutput* )

unpack DUN call info indication

## 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

**9.45.7.82** 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

**9.45.7.83** 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

**9.45.7.84** int unpack\_wds\_SLQSGetCurrentChannelRate ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSGetCurrentChannelRate\_t \* *pOutput* )

get current Tx/Rx channel bitrate of the current packet data 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

**9.45.7.85** 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

**9.45.7.86** 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

**9.45.7.87** int unpack\_wds\_SLQSGetProfileSettings ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSGetProfileSettings\_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 structure to fill

**Returns**

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

**See Also**

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

**9.45.7.88** int unpack\_wds\_SLQSGetProfileSettingsV2 ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSGetProfileSettingsV2\_t \* *pOutput* )

get profile settings unpack V2

**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

**9.45.7.89** 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

**9.45.7.90** int unpack\_wds\_SLQSModifyProfile ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSModifyProfile\_t  
\* *pOutput* )

Modify Profile unpack

**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

**9.45.7.91** int unpack\_wds\_SLQSResetPacketStatics ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSResetPacketStatics\_t \* *pOutput* )

Reset packet data transfer statistics unpack.

## 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

**9.45.7.92** int unpack\_wds\_SLQSSet3GPPConfigItem ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSet3GPPConfigItem\_t \* *pOutput* )

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>	unpacked response

**9.45.7.93** 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

9.45.7.94 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

9.45.7.95 int unpack\_wds\_SLQSSetWdsEventCallback ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSetWdsEventCallback\_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>	unpacked response

## Returns

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

## See Also

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

9.45.7.96 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

**9.45.7.97** 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

**9.45.7.98** int unpack\_wds\_SLQSSGetLoopback ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSSGetLoopback-  
\_t \* *pOutput* )

get the value of loopback mode and multiplier 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

**9.45.7.99** int unpack\_wds\_SLQSSSetDHCPv4ClientConfig ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSSetDHCPv4ClientConfig\_t \* *pOutput* )

Gets the DHCP Client V4 Configuration unpack.

## 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

**9.45.7.100** int unpack\_wds\_SLQSSSetLoopback ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSSSetLoopback\_t \* *pOutput* )

Enable/disable Data Loopback Mode and set the value of loopback multiplier unpack

**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

**9.45.7.101** int unpack\_wds\_SLQSStartDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSStartDataSession\_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

**9.45.7.102** int unpack\_wds\_SLQSStopDataSession ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSStopDataSession\_t \* *pOutput* )

stop data session unpack

**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

**9.45.7.103** int unpack\_wds\_SLQSSwiProfileChangeCallback ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSwiProfileChangeCallback\_t \* *pOutput* )

unpack profile change info

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked
out	<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

**9.45.7.104** int unpack\_wds\_SLQSSwiProfileChangeCallback\_Ind ( uint8\_t \* *pResp*, uint16\_t *respLen*,  
unpack\_wds\_SLQSSwiProfileChangeCallback\_Ind\_t \* *pOutput* )

unpack profile change indication

**Parameters**

in	<i>pResp</i>	qmi response from modem
in	<i>respLen</i>	qmi response length
out	<i>pOutput</i>	response unpacked
out	<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

**9.45.7.105** int unpack\_wds\_SLQSWdsGoActive ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSWdsGoActive\_t  
\* *pOutput* )

Gets the device into Active state unpack

## 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

9.45.7.106 int unpack\_wds\_SLQSWdsGoDormant ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSWdsGoDormant\_t \* *pOutput* )

Gets the device into dormant state unpack

## 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

9.45.7.107 int unpack\_wds\_SLQSWdsSetEventReport ( uint8\_t \* *pResp*, uint16\_t *respLen*, unpack\_wds\_SLQSWdsSetEventReport\_t \* *pOutput* )

Sets the event report parameters unpack

## 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

9.45.7.108 `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](#)
  - [unpack\\_wds\\_SLQSGet3GPPConfigItem\\_t, 1073](#)
- [\\_litew\\_FirmwareFileInfo, 35](#)
  - [carrierStr, 36](#)
  - [fullPath, 36](#)
  - [headerType, 36](#)
  - [imageMask, 36](#)
  - [imageType, 36](#)
  - [modelIdStr, 36](#)
  - [packageIdStr, 36](#)
  - [partNoStr, 36](#)
  - [priVersionStr, 36](#)
  - [releaseDate, 36](#)
  - [skuStr, 36](#)
  - [versionStr, 36](#)
- [\\_litew\\_FirmwareInfo\\_, 36](#)
  - [szCarrier\\_str, 37](#)
  - [szCarrierPriversion\\_str, 37](#)
  - [szFwversion\\_str, 37](#)
  - [szModelid\\_str, 37](#)
  - [szPackageid\\_str, 37](#)
  - [szSku\\_str, 37](#)
- [\\_litew\\_FirmwarePartNo\\_, 37](#)
  - [szPartno\\_str, 38](#)
- [AAASPI](#)
  - [unpack\\_wds\\_GetMobileIPProfile\\_t, 1062](#)
- [AAASState](#)
  - [unpack\\_wds\\_GetMobileIPProfile\\_t, 1062](#)
- [ACT\\_CODE\\_MAX\\_SIZE](#)
  - [dms.h, 1206](#)
- [ALS](#)
  - [voice\\_getAllCallInformation, 1142](#)
- [AMSSString](#)
  - [unpack\\_dms\\_GetFirmwareRevision\\_t, 679](#)
  - [unpack\\_dms\\_GetFirmwareRevisions\\_t, 681](#)
- [AMTlv](#)
  - [unpack\\_ims\\_SLQSVolPCfgCallback\\_ind\\_t, 752](#)
- [AOATlv](#)
  - [unpack\\_ims\\_SLQSVolPCfgCallback\\_ind\\_t, 752](#)
- [AOPTlv](#)
  - [NASQmiCbkNasSystemSelPrefInd, 368](#)
- [APNName](#)
  - [unpack\\_wds\\_SLQSGetRuntimeSettings\\_t, 1083](#)
- [AWMTlv](#)
  - [unpack\\_ims\\_SLQSVolPCfgCallback\\_ind\\_t, 752](#)
- [AWOATlv](#)
  - [unpack\\_ims\\_SLQSVolPCfgCallback\\_ind\\_t, 752](#)
- [accelTemp](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 419](#)
- [acceleroData](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 419](#)
- [acceleroTimeSrc](#)
  - [pack\\_loc\\_SLQSLOCInjectSensorData\\_t, 419](#)
- [AccessMac](#)
  - [nas\\_protocolSubtypeElement, 290](#)
- [accolc](#)
  - [pack\\_nas\\_SetACCOLC\\_t, 428](#)
- [accuracy](#)
  - [loc\\_HorAccuracyLvITlv, 158](#)
  - [pack\\_pds\\_SetPDSDefaults\\_t, 476](#)
- [ackIndicator](#)
  - [sMSTransferRouteMTMessageInfo, 613](#)
- [acqOrdeLen](#)
  - [nas\\_acqOrderPref, 171](#)
  - [nas\\_AcqOrderPrefTlv, 172](#)
- [AcqOrderLen](#)
  - [NASAcqOrderPrefTlv, 352](#)
- [AcqOrderPref](#)
  - [NASAcqOrderPrefTlv, 352](#)
- [acroamsetting](#)
  - [pack\\_wds\\_SetAutoconnect\\_t, 560](#)
- [acsetting](#)
  - [pack\\_wds\\_SetAutoconnect\\_t, 560](#)
- [actCode](#)
  - [pack\\_dms\\_ActivateAutomatic\\_t, 381](#)
- [ActSetCnt](#)
  - [nas\\_NetworkStat1x, 261](#)
- [ActSetPilotPN](#)
  - [nas\\_ActPilotPNElement, 173](#)
- [ActSetPilotPNStrength](#)
  - [nas\\_ActPilotPNElement, 173](#)
- [action](#)
  - [pack\\_nas\\_InitiateDomainAttach\\_t, 424](#)
- [activated\\_ind](#)
  - [sms\\_qaQmi3GPP2BroadcastCfgInfo, 602](#)
  - [sms\\_qaQmi3GPPBroadcastCfgInfo, 602](#)
- [activationState](#)
  - [pack\\_dms\\_SetActivationStatusCallback\\_t, 382](#)
- [activationStatus](#)
  - [dms\\_ActivationStatusTlv, 56](#)
- [ActivationStatusTlv](#)
  - [unpack\\_dms\\_SetEventReport\\_ind\\_t, 697](#)
- [activeBand](#)
  - [nas\\_RfBandInfoExtFormatElements, 299](#)
  - [nas\\_RFBandInfoExtTlv, 300](#)
  - [nas\\_RFBandInfoTlv, 302](#)
- [activeBandClass](#)
  - [nas\\_RFBandInfoElements, 298](#)

- nas\_RFInfoTlv, 308
- RFBandInfoElements, 590
- activeChannel
  - nas\_RFBandInfoElements, 298
  - nas\_RfBandInfoExtFormatElements, 299
  - nas\_RFBandInfoExtTlv, 301
  - nas\_RFBandInfoTlv, 302
  - nas\_RFInfoTlv, 308
  - RFBandInfoElements, 590
- activeInd
  - sms\_messageWaitingInfoContent, 600
- ActiveStatus
  - voice\_CLIPResp, 1132
  - voice\_CLIRResp, 1133
  - voice\_CNAPResp, 1134
  - voice\_COLPResp, 1134
  - voice\_COLRResp, 1135
- ActiveTechPref
  - unpack\_nas\_GetNetworkPreference\_t, 801
- ActiveTimer
  - unpack\_dms\_PSMCfgChange\_ind\_t, 694
- activeTimer
  - dms\_PSMActiveTimerTlv, 67
- ActiveTimerInd
  - dms\_PSMActiveTimerIndTlv, 66
- additionalRecord
  - uim\_additionalReadResult, 622
- additionalRecordLen
  - uim\_additionalReadResult, 622
- addr
  - unpack\_qos\_IPv4Addr\_t, 907
  - unpack\_qos\_IPv6Addr\_t, 908
- address
  - loc\_IPv4Info, 160
  - loc\_IPv6Info, 161
  - loc\_urlAddr, 169
  - unpack\_wds\_GetMobileIPProfile\_t, 1062
- addressSize
  - unpack\_sms\_GetSMSCAddress\_t, 932
- aggDIBW
  - nas\_PhyCaAggDIBW, 279
- aid
  - uim\_refreshevent, 643
  - uim\_sessionInformation, 646
  - uim\_UIMSessionInformation, 651
- aidLength
  - appStats, 41
  - uim\_appStatus, 624
  - uim\_refreshevent, 643
  - uim\_sessionInformation, 646
  - uim\_UIMSessionInformation, 652
- aidVal
  - appStats, 41
  - uim\_appStatus, 624
- aidingIndicatorMask
  - loc\_sensorDataUsage, 166
- airTimerValue
  - voice\_airTimer, 1110
- alertPitch
  - voice\_signalInfo, 1154
- alertingPattern
  - voice\_arrAlertingPattern, 1113
- AlertingType
  - voice\_arrAlertingType, 1114
- alertmsg
  - unpack\_omaDmConfigTlv\_t, 889
- alertmsglength
  - unpack\_omaDmConfigTlv\_t, 889
- allCallsAlphaIDInfoArr
  - voice\_arrAlphaID, 1114
- AllCallsUUSInfo
  - voice\_arrUUSInfo, 1120
- alphaDcs
  - voice\_alphaIDInfo, 1112
- AlphaID
  - cat\_AIPhIdIdentifierTlv, 49
- AlphaIDInfo
  - voice\_allCallsAlphaIDInfo, 1111
- alphaIDLen
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-\_ind\_t, 949
- AlphaIDLength
  - cat\_AIPhIdIdentifierTlv, 49
- alphaLen
  - voice\_alphaIDInfo, 1112
- alphaText
  - voice\_alphaIDInfo, 1112
- altSrcInfo\_t, 38
  - coverage, 39
  - linkage, 39
  - source, 39
- Altitude
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- altitudeAssumed
  - unpack\_loc\_GnssSvInfo\_Ind\_t, 776
- altitudeSrcInfo
  - pack\_loc\_SLQSLOCInjectPosition\_t, 417
- altitudeWrtEllipsoid
  - pack\_loc\_SLQSLOCInjectPosition\_t, 417
- altitudeWrtMeanSeaLevel
  - pack\_loc\_SLQSLOCInjectPosition\_t, 417
- ambr\_dl
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 913
- ambr\_dl\_ext
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 913
- ambr\_dl\_ext2
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 913
- ambr\_ul
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 913
- ambr\_ul\_ext
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams-\_t, 913

- ambr\_ul\_ext2
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [913](#)
- amrMode
  - ims\_AMRModelInfo, [84](#)
- amrOctAlign
  - ims\_AMROctAlignInfo, [85](#)
- amrWBEnable
  - ims\_EnabAMRWBInfo, [87](#)
- amrWBMode
  - ims\_AMRWBModelInfo, [86](#)
- amrWBOctAlign
  - ims\_AMRWBOctAlignInfo, [86](#)
- amssSize
  - unpack\_dms\_GetFirmwareRevision\_t, [679](#)
  - unpack\_dms\_GetFirmwareRevisions\_t, [681](#)
- apdxyPages.c, [1183](#)
- apnId
  - pack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [486](#)
  - pack\_qos\_SLQSQosSwiReadDataStats\_t, [486](#)
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [913](#)
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- apnName
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1098](#)
- apnname
  - unpack\_wds\_GetDefaultProfile\_t, [1054](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- apnsize
  - unpack\_wds\_GetDefaultProfile\_t, [1054](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- appNameLength
  - loc\_LocApplicationInfo, [162](#)
- appProviderLength
  - loc\_LocApplicationInfo, [162](#)
- appState
  - appStats, [41](#)
  - uim\_appStatus, [624](#)
- appStats, [39](#)
  - aidLength, [41](#)
  - aidVal, [41](#)
  - appState, [41](#)
  - appType, [41](#)
  - persoFeature, [41](#)
  - persoRetries, [41](#)
  - persoState, [41](#)
  - persoUnblockRetries, [41](#)
  - pin1Retries, [42](#)
  - pin1State, [42](#)
  - pin2Retries, [42](#)
  - pin2State, [42](#)
  - puk1Retries, [42](#)
  - puk2Retries, [42](#)
  - univPin, [42](#)
- AppStatus
  - slotInf, [594](#)
  - uim\_slotInfo, [649](#)
- appType
  - appStats, [41](#)
  - uim\_appStatus, [625](#)
- appVersionLength
  - loc\_LocApplicationInfo, [162](#)
- appVersionValid
  - loc\_LocApplicationInfo, [162](#)
- Application
  - unpack\_nas\_GetCDMANetworkParameters\_t, [796](#)
- appversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
- arfcn
  - nas\_GERANInfo, [206](#)
  - nas\_gsmCellInfo, [209](#)
- arrCallInformation
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
- arrfileInfo
  - uim\_refreshevent, [643](#)
  - uim\_registerRefresh, [644](#)
- atrValue
  - uim\_physlotInfo, [639](#)
- atrValueLen
  - uim\_physlotInfo, [639](#)
- audio.h, [1183](#)
  - pack\_audio\_SLQSGetAudioPathConfig, [1184](#)
  - pack\_audio\_SLQSGetAudioProfile, [1184](#)
  - pack\_audio\_SLQSGetAudioVolTLBConfig, [1185](#)
  - pack\_audio\_SLQSSetAudioPathConfig, [1185](#)
  - pack\_audio\_SLQSSetAudioProfile, [1186](#)
  - pack\_audio\_SLQSSetAudioVolTLBConfig, [1186](#)
  - unpack\_audio\_SLQSGetAudioPathConfig, [1187](#)
  - unpack\_audio\_SLQSGetAudioProfile, [1187](#)
  - unpack\_audio\_SLQSGetAudioVolTLBConfig, [1187](#)
  - unpack\_audio\_SLQSSetAudioPathConfig, [1188](#)
  - unpack\_audio\_SLQSSetAudioProfile, [1188](#)
  - unpack\_audio\_SLQSSetAudioVolTLBConfig, [1188](#)
- audio\_RXAGCList, [42](#)
  - pRXAIG, [42](#)
  - pRXComprSlope, [43](#)
  - pRXComprThres, [43](#)
  - pRXExpSlope, [43](#)
  - pRXExpThres, [43](#)
  - pRXStaticGain, [43](#)
- audio\_RXAVCList, [43](#)
  - pAVRXAVCHadroom, [43](#)
  - pAVRXAVCSens, [43](#)
- audio\_RXPCMIRFitr, [43](#)
  - pFlag, [45](#)
  - pStage0Val, [45](#)
  - pStage1Val, [45](#)
  - pStage2Val, [45](#)
  - pStage3Val, [45](#)
  - pStage4Val, [45](#)
  - pStageCnt, [45](#)
- audio\_TXAGCList, [45](#)
  - pTXAIG, [45](#)
  - pTXComprSlope, [46](#)



- pTXComprThres, [46](#)
  - pTXExpSlope, [46](#)
  - pTXExpThres, [46](#)
  - pTXStaticGain, [46](#)
- audio\_TXPCMIIRFtr, [46](#)
- pFlag, [47](#)
  - pStage0Val, [47](#)
  - pStage1Val, [47](#)
  - pStage2Val, [47](#)
  - pStage3Val, [47](#)
  - pStage4Val, [47](#)
  - pStageCnt, [47](#)
- auth
  - unpack\_wds\_GetDefaultProfile\_t, [1054](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- authData
  - pack\_uim\_SLQSUIAuthenticate\_t, [524](#)
- AuthProt
  - nas\_protocolSubtypeElement, [290](#)
- Authentication
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1083](#)
- authentication
  - pack\_wds\_SetDefaultProfile\_t, [562](#)
- AutoConnect
  - pack\_swiavms\_SLQSAVMSSetSettings\_t, [506](#)
  - pack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, [507](#)
  - unpack\_swiavms\_SLQSAVMSSetSettings\_t, [957](#)
  - unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, [960](#)
- AutoReboot
  - pack\_swiavms\_SLQSAVMSSetSettings\_t, [506](#)
  - unpack\_swiavms\_SLQSAVMSSetSettings\_t, [957](#)
- Autosdm
  - unpack\_swioma\_SLQSOMADMGetSettings\_t, [980](#)
- avgPeriod
  - nas\_LTESigRptCfg, [247](#)
  - nas\_LTESigRptConfig, [248](#)
- azimuth
  - loc\_satelliteInfo, [165](#)
- bAPNLength
  - PackSwiAvmsSetSettingsAPNInfo, [583](#)
  - PackSwiAVMSSettingsAPNInfo, [585](#)
- bAuto
  - pack\_pds\_SetPortAutomaticTracking\_t, [477](#)
  - pack\_pds\_SetServiceAutomaticTracking\_t, [477](#)
- bBinaryType
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1101](#)
- bEnable
  - pack\_dms\_UIMSetPINProtection\_t, [397](#)
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t, [457](#)
- bEnabled
  - pack\_pds\_SetXTRAAutomaticDownload\_t, [478](#)
- bForceDownload
  - pack\_fms\_SetImagesPreference\_t, [400](#)
- bICCID
  - slot\_t, [593](#)
- bICCIDLength
  - slot\_t, [593](#)
- bLogicalSlot
  - pack\_uim\_SLQSUIMSwitchSlot\_t, [532](#)
  - slot\_t, [593](#)
- bNotification
  - UnpackSwiAvmsEventReportNotification, [1106](#)
- bNumberOfPhySlots
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, [989](#)
- BOOL
  - SwiDataTypes.h, [1517](#)
- BPTlv
  - NASQmiCbkNasSystemSelPrefInd, [368](#)
- bPWDLength
  - PackSwiAvmsSetSettingsAPNInfo, [583](#)
  - PackSwiAVMSSettingsAPNInfo, [585](#)
- bPackageID
  - UnpackSwiAvmsEventReportPackageID, [1107](#)
- bRegStatus
  - UnpackSwiAvmsEventReportRegStatus, [1107](#)
- bResetStatistics
  - rmTrasnferStaticsReq, [590](#)
- bSerity
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1101](#)
- bState
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1101](#)
  - UnpackSwiAvmsEventReportConfig, [1103](#)
- bType
  - UnpackSwiAvmsEventReportDataSessionStatus, [1104](#)
  - UnpackSwiAvmsEventReportSessionType, [1108](#)
- bUnameLength
  - PackSwiAvmsSetSettingsAPNInfo, [583](#)
  - PackSwiAVMSSettingsAPNInfo, [585](#)
- bUserInputRequest
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1101](#)
  - UnpackSwiAvmsEventReportConfig, [1103](#)
  - UnpackSwiAvmsEventReportConnectionRequest, [1103](#)
- BYT\_STAT\_STAT\_MASK
  - wds.h, [1614](#)
- BYTE
  - SwiDataTypes.h, [1517](#)
- band
  - nas\_LTEInfo, [231](#)
  - NASPhyCaAggScellArray, [362](#)
- band1900
  - nas\_gsmCellInfo, [209](#)
- band\_pref
  - NASBandPreferenceTlv, [353](#)
- BandCapability
  - unpack\_dms\_GetBandCapability\_t, [664](#)
- bandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, [711](#)

- unpack\_dms\_SLQSGetBandCapabilityExt\_t, 715
- BandPref
  - nas\_BandPrefTlv, 177
- bandPref
  - pack\_nas\_SLQSSetBandPreference\_t, 456
- bandwidth
  - nas\_LTEInfo, 231
  - nas\_RfBandwidthInfoElements, 304
  - nas\_RFBandwidthInfoTlv, 305
- baseId
  - nas\_CDMAInfo, 180
  - nas\_CDMASysInfo, 184
- baseLat
  - nas\_CDMAInfo, 180
  - nas\_CDMASysInfo, 184
- baseLong
  - nas\_CDMAInfo, 180
  - nas\_CDMASysInfo, 184
- BasestationID
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- BasestationLatitude
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- BasestationLongitude
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- batchPerSec
  - loc\_accelAcceptReady, 148
  - loc\_accelTempAcceptReady, 149
  - loc\_gyroAcceptReady, 157
  - loc\_gyroTempAcceptReady, 157
- BearerID
  - unpack\_qos\_QosFlowInfo\_t, 910
- bearerID
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 1087
- bearerId
  - unpack\_QosFlowStat\_t, 928
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- bits\_129\_192
  - nas\_BandPrefInfoTlv, 174
  - nas\_lteBandPrefExt, 224
  - nas\_nr5gBandPref, 265
- bits\_193\_256
  - nas\_BandPrefInfoTlv, 174
  - nas\_lteBandPrefExt, 224
  - nas\_nr5gBandPref, 265
- bits\_1\_64
  - nas\_BandPrefInfoTlv, 174
  - nas\_lteBandPrefExt, 224
  - nas\_nr5gBandPref, 265
- bits\_65\_128
  - nas\_BandPrefInfoTlv, 175
  - nas\_lteBandPrefExt, 224
  - nas\_nr5gBandPref, 265
- bootSize
  - unpack\_dms\_GetFirmwareRevisions\_t, 681
- BootString
  - unpack\_dms\_GetFirmwareRevisions\_t, 681
- bootversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 678
- Broadcast
  - unpack\_nas\_GetCDMANetworkParameters\_t, 796
- broadcastActivate
  - pack\_sms\_SLQSSetSmsBroadcastActivation\_t, 495
- broadcastConfig
  - sms\_qaQmi3GPPBroadcastCfgInfo, 602
- bsInfoValid
  - nas\_CDMASysInfo, 184
- bsPRev
  - nas\_CDMASysInfo, 184
- bsPRevValid
  - nas\_CDMASysInfo, 184
- bsic
  - nas\_GERANInfo, 206
- bsicId
  - nas\_gsmCellInfo, 209
- bucketSz
  - unpack\_qos\_tokenBucket\_t, 927
- buildID
  - FMSImageldElement, 80
  - image\_info\_t, 84
- buildIDLen
  - image\_info\_t, 84
- buildIDLength
  - FMSImageldElement, 80
- buildId
  - FMSImageElement, 79
- buildIdLength
  - FMSImageElement, 79
- BurstDTMFInfo
  - pack\_voice\_SLQSVoiceBurstDTMF\_t, 539
- ByteLoopbackMode
  - unpack\_wds\_SLQSSetLoopback\_t, 1095
- ByteLoopbackMultiplier
  - unpack\_wds\_SLQSSetLoopback\_t, 1095
- CCETlv
  - unpack\_cat\_SetCatEventCallback\_ind\_t, 660
- CDMA\_P\_Rev
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- CDMABroadcastConfig
  - sms\_qaQmi3GPP2BroadcastCfgInfo, 602
- CDMAECIOThreshListLen
  - nas\_CDMAECIOThresh, 179
- CDMARSSIThreshListLen
  - nas\_CDMARSSIThresh, 181
- CDMASSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 865
- CDMASystemInfoExt
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- CHAR

- SwiDataTypes.h, [1517](#)
- CK\_MAX\_SIZE
  - dms.h, [1206](#)
- CQIValueCW0
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, [882](#)
- CQIValueCW1
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, [882](#)
- CRTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, [1072](#)
- CSDomain
  - unpack\_nas\_GetServingNetwork\_t, [804](#)
- CSTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, [1072](#)
- CUGIndex
  - voice\_CUGInfo, [1137](#)
- CallBarStatus
  - unpack\_nas\_SLQSGetServingSystem\_t, [825](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- callDuration
  - unpack\_wds\_GetSessionDuration\_t, [1067](#)
  - unpack\_wds\_GetSessionDurationV2\_t, [1068](#)
- CallEndReason
  - wds\_LastMdmCallEndRsnTlv, [1171](#)
- callEndReason
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [1079](#)
  - voice\_arrCallEndReason, [1116](#)
- CallFWExtInfo
  - voice\_getCallFWExtInfo, [1144](#)
- CallFWInfo
  - voice\_getCallFWInfo, [1144](#)
- callID
  - pack\_voice\_SLQSVoiceGetCallInfo\_t, [544](#)
  - pack\_voice\_SLQSVoiceStopContDTMF\_t, [555](#)
  - unpack\_voice\_OTASPStatusCallback\_ind\_t, [1006](#)
  - unpack\_voice\_SLQSVoiceStopContDTMF\_t, [1039](#)
  - unpack\_voice\_SUPSNotificationCallback\_ind\_t, [1044](#)
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
  - unpack\_voice\_voicePrivacyChangeCallback\_ind\_t, [1048](#)
  - voice\_allCallsAlphaIDInfo, [1111](#)
  - voice\_allCallsDiagInfo, [1111](#)
  - voice\_allCallsUUSInfo, [1112](#)
  - voice\_arrAlertingPattern, [1113](#)
  - voice\_arrAlertingType, [1114](#)
  - voice\_arrCallEndReason, [1116](#)
  - voice\_arrSvcOption, [1119](#)
  - voice\_callInfo, [1129](#)
  - voice\_DTMFInfo, [1139](#)
  - voice\_getAllCallRmtPtyName, [1143](#)
  - voice\_getAllCallRmtPtyNum, [1143](#)
  - voice\_peerNumberInfo, [1147](#)
- callNumber
  - pack\_voice\_SLQSVoiceDialCall\_t, [541](#)
- callState
  - voice\_callInfo, [1129](#)
- callType
  - voice\_callInfo, [1129](#)
- CalledPartyNum
  - voice\_arrCalledPartyNum, [1115](#)
- callerID
  - voice\_callerIDInfo, [1124](#)
  - voice\_connectNumInfo, [1136](#)
- callerIDLen
  - voice\_callerIDInfo, [1124](#)
  - voice\_connectNumInfo, [1136](#)
- callerName
  - voice\_remotePartyName, [1152](#)
- Callinfo
  - voice\_getAllCallInformation, [1142](#)
- campedCiotLteOpMode
  - nas\_LteCiotOpModeTlv, [226](#)
- capMask
  - pack\_uim\_SLQSUIMGetServiceStatus\_t, [527](#)
- cardProtocol
  - uim\_physlotInfo, [639](#)
- cardState
  - slotInf, [594](#)
  - uim\_slotInfo, [649](#)
- carrier
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [721](#)
- carrier\_str
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
- CarrierImage\_t, [47](#)
  - m\_FwBuildId, [48](#)
  - m\_FwImgId, [48](#)
  - m\_PriBuildId, [48](#)
  - m\_PriImgId, [49](#)
  - m\_nCarrierId, [48](#)
  - m\_nFolderId, [48](#)
  - m\_nStorage, [48](#)
- carrierStr
  - \_litefw\_FirmwareFileInfo, [36](#)
- cat.h, [1189](#)
  - pack\_cat\_CATSendEnvelopeCommand, [1190](#)
  - pack\_cat\_CATSendTerminalResponse, [1190](#)
  - pack\_cat\_SetCATEventCallback, [1190](#)
  - unpack\_cat\_CATSendEnvelopeCommand, [1191](#)
  - unpack\_cat\_CATSendEnvelopeCommand\_t, [1190](#)
  - unpack\_cat\_CATSendTerminalResponse, [1191](#)
  - unpack\_cat\_CATSendTerminalResponse\_t, [1190](#)
  - unpack\_cat\_SetCATEventCallback, [1192](#)
  - unpack\_cat\_SetCatEventCallback\_ind, [1192](#)
- cat\_AlphaIdentifierTlv, [49](#)
  - AlphaID, [49](#)
  - AlphaIDLength, [49](#)
  - ReferenceID, [49](#)
- cat\_EndProactiveSessionTlv, [51](#)
  - EndProactiveSession, [51](#)
- cat\_EventIDDDataTlv, [51](#)
  - Data, [51](#)
  - DataLength, [51](#)
  - ReferenceID, [51](#)
- cat\_EventListTlv, [51](#)

- SetupEventList, 52
- cat\_RefreshTlv, 52
  - RefreshMode, 52
  - RefreshStage, 52
- cat\_commonEventTlv, 49
  - CatEvent, 50
  - EventID, 50
  - EventLength, 50
  - TlvPresent, 50
- cat\_currentCatEvent, 50
  - CatAlphaDtf, 50
  - CatEndPS, 50
  - CatEvIDData, 50
  - CatEventLst, 50
  - CatRefresh, 50
- CatAlphaDtf
  - cat\_currentCatEvent, 50
- CatEndPS
  - cat\_currentCatEvent, 50
- CatEvIDData
  - cat\_currentCatEvent, 50
- CatEvent
  - cat\_commonEventTlv, 50
- CatEventLst
  - cat\_currentCatEvent, 50
- CatRefresh
  - cat\_currentCatEvent, 50
- causeCode
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-  
\_ind\_t, 949
- ccsSupported
  - nas\_CDMA SysInfo, 184
- ccsSupportedValid
  - nas\_CDMA SysInfo, 184
- cdmaSSInfo, 52
  - ecio, 53
  - rssI, 53
- cdmaSysIdValid
  - nas\_CDMA SysInfo, 184
- cell\_resel\_priority
  - nas\_infoInterFreq, 222
- cellBroadcastCap
  - nas\_AddSysInfo, 174
- CellID
  - unpack\_nas\_SLQSGetServingSystem\_t, 825
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- cellID
  - nas\_GERANInfo, 206
  - nas\_QmisNasSlqsNasPCICellInfo, 294
  - nas\_UMTSInfo, 339
- cellId
  - nas\_GSM SysInfo, 213
  - nas\_LTE SysInfo, 251
  - nas\_NR5GSysInfoTlv, 271
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_UMTSExtInfo, 337
  - nas\_WCDMA SysInfo, 351
- cellIdValid
  - nas\_gsmCellInfo, 209
  - nas\_GSM SysInfo, 213
  - nas\_LTE SysInfo, 252
  - nas\_NR5GSysInfoTlv, 271
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_WCDMA SysInfo, 351
- cellInterFreqParams
  - nas\_infoInterFreq, 222
- cellsTDD
  - nas\_umtsLTENbrCell, 342
- CellParams
  - nas\_LTEInfoIntraFreq, 234
- cellReselPriority
  - nas\_lteGsmCellInfo, 229
  - nas\_LTEInfoIntraFreq, 234
  - nas\_lteWcdmaCellInfo, 253
- cells\_len
  - nas\_infoInterFreq, 222
  - nas\_lteGsmCellInfo, 229
- cellsLen
  - nas\_LTEInfoIntraFreq, 234
  - nas\_lteWcdmaCellInfo, 253
- CfgValue
  - pack\_swidms\_SLQSSwiDmsSetUsbComp\_t, 511
  - swidms\_ifaceCfgTlv, 616
- chaddr
  - wds\_DHCPv4HWConfig, 1165
  - wdsDhcpv4HwConfig, 1180
- chaddrLen
  - wds\_DHCPv4HWConfig, 1165
  - wdsDhcpv4HwConfig, 1180
- changePIN
  - pack\_uim\_ChangePin\_t, 521
- ChannelRate
  - wds\_ChannelRateTlv, 1158
- channelRate
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1079
- Chipset
  - nas\_DeviceConfigDetail, 197
- CiotAOPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- CiotAcqOrderLen
  - NASCiotAcqOrderPrefTlv, 353
- ciotAcqOrderLen
  - nas\_ciotAcqOrderPref, 187
  - nas\_CiotAcqOrderPrefTlv, 188
- CiotAcqOrderPref
  - NASCiotAcqOrderPrefTlv, 353
- ciotLteOpMode
  - nas\_LTEOperationalModeTlv, 240
- CiotLteOpModePref
  - NASCiotLteOpModePrefTlv, 354
- ciotLteOpModePref
  - nas\_CiotLteOpModePrefTlv, 188
- CiotOpMPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- ckLen
  - uim\_depersonalizationInformation, 631

- ckVal
  - uim\_depersonalizationInformation, [631](#)
- clear
  - pack\_dms\_SLQSSwiGetCrashInfo\_t, [390](#)
- cmdID
  - pack\_cat\_CATSendEnvelopeCommand\_t, [379](#)
- codingScheme
  - nas\_PLMNNetworkNameData, [288](#)
  - voice\_remotePartyName, [1152](#)
- common.h
  - eAUDIO, [1196](#)
  - eCAT, [1195](#)
  - eCTL, [1195](#)
  - eDMS, [1195](#)
  - eIMS, [1195](#)
  - eIMSA, [1195](#)
  - eIND, [1196](#)
  - eLOC, [1195](#)
  - eLOG\_DEBUG, [1195](#)
  - eLOG\_FATAL, [1195](#)
  - eLOG\_INFO, [1195](#)
  - eLOG\_WARN, [1195](#)
  - eNAS, [1195](#)
  - ePDS, [1195](#)
  - eQOS, [1195](#)
  - eREQ, [1196](#)
  - eRMS, [1195](#)
  - eRSP, [1196](#)
  - eSAR, [1195](#)
  - eSMS, [1195](#)
  - eSWIAUDIO, [1196](#)
  - eSWIDMS, [1196](#)
  - eSWILOC, [1196](#)
  - eSWIM2MCMD, [1196](#)
  - eSWIM2MCMD\_AVC2, [1196](#)
  - eSWIOMA, [1195](#)
  - eSWIOMAEXT, [1196](#)
  - eTIMEOUT\_10\_S, [1196](#)
  - eTIMEOUT\_20\_S, [1196](#)
  - eTIMEOUT\_2\_S, [1196](#)
  - eTIMEOUT\_300\_S, [1196](#)
  - eTIMEOUT\_30\_S, [1196](#)
  - eTIMEOUT\_5\_S, [1196](#)
  - eTIMEOUT\_60\_S, [1196](#)
  - eTIMEOUT\_8\_S, [1196](#)
  - eTIMEOUT\_DEFAULT, [1196](#)
  - eTMD, [1195](#)
  - eUIM, [1195](#)
  - eVOICE, [1195](#)
  - eWDS, [1195](#)
- common.h, [1192](#)
  - eLOG\_LEVEL, [1195](#)
  - eQMI\_SVC, [1195](#)
  - eTimeout, [1196](#)
  - fill\_pack\_ctx, [1196](#)
  - fill\_sdu\_hdr, [1196](#)
  - get\_version, [1196](#)
  - glog, [1198](#)
  - gloglvl, [1198](#)
  - helper\_get\_error\_code, [1196](#)
  - helper\_get\_error\_reason, [1197](#)
  - helper\_get\_req\_str, [1197](#)
  - helper\_get\_resp\_ctx, [1197](#)
  - helper\_get\_xid, [1197](#)
  - helper\_isBootLoader\_DebugEnabled, [1197](#)
  - helper\_set\_log\_func, [1197](#)
  - helper\_set\_log\_lvl, [1198](#)
  - liteqmi\_GetVersion, [1198](#)
  - liteqmi\_helper\_decode7bitAsciiEncString, [1198](#)
  - liteqmi\_log, [1198](#)
  - MINREQBKLEN, [1194](#)
  - MSGID\_AND\_LEN, [1194](#)
  - MSGID\_DONT\_CARE, [1194](#)
  - msgtype, [1196](#)
  - SDU\_HDR\_LEN, [1195](#)
  - UNUSEDPARAM, [1195](#)
  - unpack\_result\_code\_only, [1198](#)
- commonInfo
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, [874](#)
- ConcSvcInfo
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- confidence
  - pack\_loc\_SLQSLOCSetCradleMountConfig\_t, [421](#)
- conn\_status
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [1087](#)
- ConnectedPartyNum
  - voice\_arrConnectPartyNum, [1117](#)
- connectionStatus, [53](#)
  - MDMCallDuration, [53](#)
  - MDMConnStatus, [53](#)
  - unpack\_wds\_GetSessionState\_t, [1068](#)
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [1079](#)
- content
  - uim\_authenticateResult, [625](#)
  - uim\_readResult, [641](#)
  - uim\_readResultInfo, [641](#)
- contentLen
  - uim\_authenticateResult, [625](#)
  - uim\_readResult, [641](#)
  - uim\_readResultInfo, [641](#)
- context
  - uim\_authenticationData, [626](#)
- contextId
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [581](#)
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1098](#)
- contextType
  - pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [581](#)
- ControlMac
  - nas\_protocolSubtypeElement, [290](#)
- count
  - swidms\_SwiDmsGetHWWatchdog, [618](#)

- count0
  - nas\_RankIndicatorTlv, [296](#)
- count1
  - nas\_RankIndicatorTlv, [296](#)
- countryInitials
  - nas\_PLMNNetworkNameData, [288](#)
- coverage
  - altSrcInfo\_t, [39](#)
- cphy\_ca\_dl\_bandwidth
  - NASPhyCaAggScellArray, [362](#)
- cphy\_scell\_info\_list\_len
  - NASPhyCaAggScellArray, [362](#)
- cpich\_ecno
  - nas\_wcdmaCellInfo, [345](#)
- cpich\_rscp
  - nas\_wcdmaCellInfo, [345](#)
- cradleMountConfigStatus
  - unpack\_loc\_CradleMountCallback\_Ind\_t, [767](#)
- crashAction
  - pack\_dms\_SetCrashAction\_t, [383](#)
- crashData
  - crashInformation, [55](#)
- crashId
  - crashInformation, [55](#)
- crashInfo
  - crashInfoParams, [54](#)
- crashInfoParam
  - unpack\_dms\_SLQSSwiGetCrashInfo\_t, [719](#)
- crashInfoParams, [54](#)
  - crashInfo, [54](#)
  - crashStatus, [54](#)
- crashInformation, [54](#)
  - crashData, [55](#)
  - crashId, [55](#)
  - crashString, [55](#)
  - crashStrlen, [55](#)
  - gcdumpString, [55](#)
  - gcdumpStrlen, [55](#)
  - numCrashes, [55](#)
- crashStatus
  - crashInfoParams, [54](#)
- crashString
  - crashInformation, [55](#)
- crashStrlen
  - crashInformation, [55](#)
- csAttachState
  - nas\_servSystem, [319](#)
  - NASServingSystemInfo, [371](#)
- csBarStatus
  - nas\_CallBarringSysInfo, [177](#)
  - nas\_callBarStatus, [178](#)
- cscfPortName
  - ims\_CSCFPortNameInfo, [87](#)
- csgId
  - nas\_CsgId, [190](#)
- cur\_carr\_name
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
- cur\_carr\_rev
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
- curDataBearerTechnology
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, [1076](#)
- CurIndex
  - dms\_devMaxCfgListCaps, [59](#)
- curProfile
  - pack\_wds\_SLQSModifyProfile\_t, [572](#)
  - UnPackGetProfileSettingOut, [1099](#)
  - UnPackGetProfileSettingOutV2, [1100](#)
- CurSubsCapsLen
  - dms\_devCurSubsCaps, [58](#)
- CurrChanRxRate
  - dunchannelRate, [77](#)
  - wds\_channelRate, [1158](#)
- CurrChanTxRate
  - dunchannelRate, [77](#)
  - wds\_channelRate, [1158](#)
- currDBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1092](#)
- currNWInfo
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1092](#)
- currNetworkInfo, [55](#)
  - NetworkType, [56](#)
  - RATMask, [56](#)
  - SOMask, [56](#)
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, [1074](#)
- current\_channel\_rx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_t, [1076](#)
- current\_channel\_tx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_t, [1076](#)
- CurrentCfgType
  - swidms\_intfaceCfgTlv, [616](#)
- currentChannelRXRate
  - unpack\_wds\_GetConnectionRate\_t, [1051](#)
- currentChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, [1051](#)
- currentDataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [575](#)
- currentNetwork
  - qmiWDSDataBearerTechnology, [587](#)
- CurrentPLMN
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- CurrentmitigationLvl
  - unpack\_tmd\_SLQSTmdGetMitigationLvl\_t, [983](#)
- cust\_attr
  - DMScustSettingInfo, [75](#)
- cust\_id
  - DMScustSettingInfo, [75](#)
  - DMSgetCustomInput, [77](#)
  - pack\_dms\_GetCustFeaturesV2\_t, [381](#)
  - pack\_dms\_SetCustFeaturesV2\_t, [386](#)



- cust\_value
  - DMScustSettingInfo, [75](#)
  - pack\_dms\_SetCustFeaturesV2\_t, [386](#)
- custSetting
  - DMScustSettingList, [75](#)
- CustomSCP
  - unpack\_nas\_GetCDMANetworkParameters\_t, [796](#)
- CwtMute
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [951](#)
  - unpack\_swiaudio\_SLQSGetM2MAVMute\_t, [953](#)
- cycleLength
  - nas\_EdrxCycleLength, [200](#)
- DBTTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, [1072](#)
- dBTechAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1092](#)
- dBTechExtRatValue
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- dBTechExtSoMask
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- dBTechnology
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- dBtechExtAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1092](#)
- dBtechnologyExt
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- DHCPRelayEnabled
  - pack\_dms\_SetCustFeature\_t, [385](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
- DHCPv4LeaseOptTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, [1049](#)
- DHCPv4LeaseStateTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, [1049](#)
- dLatitude
  - unpack\_pds\_SetEventReport\_Ind\_t, [899](#)
- dLongitude
  - unpack\_pds\_SetEventReport\_Ind\_t, [899](#)
- DMS\_IMGDETAILS\_LEN
  - dms.h, [1206](#)
- DMS\_PM\_FACTORY
  - dms.h, [1206](#)
- DMS\_PM\_LOW
  - dms.h, [1207](#)
- DMS\_PM\_OFFLINE
  - dms.h, [1207](#)
- DMS\_PM\_ONLINE
  - dms.h, [1207](#)
- DMS\_PM\_RESET
  - dms.h, [1207](#)
- DMS\_PM\_SHUT\_DOWN
  - dms.h, [1207](#)
- DMS\_VALID\_FSN\_LEN
  - dms.h, [1207](#)
- DMScustSettingInfo, [74](#)
  - cust\_attr, [75](#)
  - cust\_id, [75](#)
  - cust\_value, [75](#)
  - id\_length, [75](#)
  - value\_length, [75](#)
- DMScustSettingList, [75](#)
  - custSetting, [75](#)
  - list\_type, [75](#)
  - num\_instances, [75](#)
- DMSgetCustomFeatureV2, [76](#)
  - pCustSettingInfo, [76](#)
  - pCustSettingList, [76](#)
  - pGetCustomInput, [76](#)
- DMSgetCustomInput, [76](#)
  - cust\_id, [77](#)
  - list\_type, [77](#)
- DRCCover
  - nas\_DRCCParams, [198](#)
- DRCValue
  - nas\_DRCCParams, [198](#)
- DSTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, [1072](#)
- DTMFEvent
  - voice\_DTMFInfo, [1139](#)
- DTMFInformation
  - unpack\_voice\_DTMFEventCallback\_ind\_t, [1004](#)
- DTMFInterdigitInterval
  - voice\_DTMFLengths, [1140](#)
- DTMFPulseWidth
  - voice\_DTMFLengths, [1140](#)
- DTMFDigit
  - pack\_voice\_SLQSVoiceStartContDTMF\_t, [554](#)
- DTMInd
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- Data
  - cat\_EventIDDDataTlv, [51](#)
- data
  - sMSCAddressInfo, [608](#)
  - sMSEtwsMessageInfo, [609](#)
  - sMSTransferRouteMTMessageInfo, [613](#)
  - uim\_authenticationData, [626](#)
  - uim\_writeRecordInfo, [654](#)
  - uim\_writeTransparentInfo, [655](#)
- data\_buf
  - NASOTAMessageTlv, [360](#)
- data\_len
  - NASOTAMessageTlv, [360](#)
- dataBearer
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [575](#)
- dataBearerMask
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, [1076](#)

- DataBearerTech
  - wds\_DataBearTechTlv, 1161
- dataBearerTech
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1079
- dataBearerTechExt
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 575
- dataCapabilities
  - nas\_dataSrvCapabilities, 193
- dataCapabilitiesLen
  - nas\_dataSrvCapabilities, 193
- DataCaps
  - unpack\_nas\_GetServingNetwork\_t, 804
  - unpack\_nas\_GetServingNetworkCapabilities\_t, 806
- dataCaps
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 809
- DataCapsLen
  - unpack\_nas\_GetServingNetwork\_t, 804
  - unpack\_nas\_GetServingNetworkCapabilities\_t, 806
- dataCapsSize
  - unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 809
- dataLen
  - pack\_cat\_CATSendEnvelopeCommand\_t, 379
  - pack\_cat\_CATSendTerminalResponse\_t, 380
  - uim\_authenticationData, 626
  - uim\_writeRecordInfo, 654
  - uim\_writeTransparentInfo, 655
- DataLength
  - cat\_EventIDDDataTlv, 51
- DataRate
  - unpack\_qos\_swiQosFlow\_t, 925
- dataRateMax
  - unpack\_qos\_dataRate\_t, 907
- dataServiceCaCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, 670
- DataServiceCapability
  - dms\_devCaps, 57
  - unpack\_dms\_GetDeviceCap\_t, 669
- DataSrvCapabilities
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- dataSysStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1092
- dataSystemStatus
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 575
- Date
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 974
- DateLength
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 974
- day
  - nas\_timeInfo, 334
  - nas\_UniversalTime, 343
- dayLtSavingAdj
  - nas\_timeInfo, 334
- dayOfWeek
  - nas\_timeInfo, 334
  - nas\_UniversalTime, 343
- daylightSavings
  - nas\_qaQmi3Gpp2TimeZone, 291
- dedicatedBand
  - nas\_RfDedicatedBandInfoElements, 307
  - nas\_RFDedicatedBandInfoTlv, 307
- defaultPDNEnabled
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1073
- DefaultRoamInd
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- delayClass
  - LibPackGPRSRequestedQoS, 105
  - wds\_GPRSQoS, 1169
- deliveryErrSDU
  - LibPackUMTSQoS, 146
  - wds\_UMTSMInQoS, 1179
- depersonalisationInfo
  - pack\_uim\_SLQSUIMDepersonalization\_t, 525
- description
  - unpack\_omaDmFotaTlv\_t, 891
- descriptionlength
  - unpack\_omaDmFotaTlv\_t, 891
- Desription
  - nas\_QmiNas3GppNetworkInfo, 292
- destPortRangeEnd
  - LibPackTFTIDParams, 143
- destPortRangeStart
  - LibPackTFTIDParams, 143
- DetailedSvcInfo
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- DevCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
- DevCfgListLen
  - dms\_devMaxCfgListCaps, 59
- DevCrashState
  - unpack\_dms\_GetCrashAction\_t, 664
- Device
  - pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, 500
- DeviceError
  - qmTlvResult, 588
- deviceId
  - unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t, 983
- deviceIdLen
  - unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t, 983
- DeviceResult
  - qmTlvResult, 588
- DiagInfo
  - voice\_allCallsDiagInfo, 1111
  - voice\_arrDiagInfo, 1117
- diagInfoLen



- voice\_diagInfo, [1138](#)
- diagnosticInfo
  - voice\_diagInfo, [1138](#)
- digitBuff
  - voice\_DTMFInfo, [1139](#)
- digitCnt
  - voice\_burstDTMFInfo, [1121](#)
  - voice\_DTMFInfo, [1139](#)
- dirNum
  - nas\_dirNum, [197](#)
- dirNumLen
  - nas\_dirNum, [197](#)
- direction
  - voice\_callInfo, [1129](#)
- DisableIMSI
  - pack\_dms\_SetCustFeature\_t, [385](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
- dispType
  - voice\_extDispRecInfo, [1141](#)
- displayCondition
  - nas\_serviceProviderName, [317](#)
- dl\_bw\_value
  - nas\_PhyCaAggPcellInfo, [280](#)
  - nas\_PhyCaAggScellIDBw, [281](#)
  - nas\_PhyCaAggScellInfo, [284](#)
  - NASPhyCaAggPcellInfo, [361](#)
  - NASPhyCaAggScellIDBw, [363](#)
  - NASPhyCaAggScellInfo, [365](#)
- dms.h, [1198](#)
  - ACT\_CODE\_MAX\_SIZE, [1206](#)
  - CK\_MAX\_SIZE, [1206](#)
  - DMS\_IMGDETAILS\_LEN, [1206](#)
  - DMS\_PM\_FACTORY, [1206](#)
  - DMS\_PM\_LOW, [1207](#)
  - DMS\_PM\_OFFLINE, [1207](#)
  - DMS\_PM\_ONLINE, [1207](#)
  - DMS\_PM\_RESET, [1207](#)
  - DMS\_PM\_SHUT\_DOWN, [1207](#)
  - DMS\_VALID\_FSN\_LEN, [1207](#)
  - ERI\_DATA\_MAX\_SIZE, [1207](#)
  - MAX\_BUILD\_ID\_LEN, [1207](#)
  - MEID\_MAX\_SIZE, [1207](#)
  - pack\_dms\_ActivateAutomatic, [1208](#)
  - pack\_dms\_GetActivationState, [1208](#)
  - pack\_dms\_GetBandCapability, [1208](#)
  - pack\_dms\_GetCrashAction, [1209](#)
  - pack\_dms\_GetCustFeature, [1209](#)
  - pack\_dms\_GetCustFeaturesV2, [1209](#)
  - pack\_dms\_GetDeviceCap, [1210](#)
  - pack\_dms\_GetDeviceCapabilities, [1210](#)
  - pack\_dms\_GetDeviceCapabilitiesV2, [1211](#)
  - pack\_dms\_GetDeviceHardwareRev, [1211](#)
  - pack\_dms\_GetDeviceMfr, [1211](#)
  - pack\_dms\_GetDeviceSerialNumbers, [1212](#)
  - pack\_dms\_GetFSN, [1213](#)
  - pack\_dms\_GetFirmwareInfo, [1212](#)
  - pack\_dms\_GetFirmwareRevision, [1212](#)
  - pack\_dms\_GetFirmwareRevisions, [1213](#)
  - pack\_dms\_GetHardwareRevision, [1213](#)
  - pack\_dms\_GetIMSI, [1214](#)
  - pack\_dms\_GetManufacturer, [1214](#)
  - pack\_dms\_GetModelID, [1214](#)
  - pack\_dms\_GetNetworkTime, [1215](#)
  - pack\_dms\_GetNetworkTimeV2, [1215](#)
  - pack\_dms\_GetOfflineReason, [1215](#)
  - pack\_dms\_GetPRLVersion, [1216](#)
  - pack\_dms\_GetPower, [1216](#)
  - pack\_dms\_GetSerialNumbers, [1216](#)
  - pack\_dms\_GetUSBComp, [1217](#)
  - pack\_dms\_GetVoiceNumber, [1217](#)
  - pack\_dms\_ResetToFactoryDefaults, [1218](#)
  - pack\_dms\_SLQSDmsSwiGetPCInfo, [1221](#)
  - pack\_dms\_SLQSDmsSwiGetResetInfo, [1222](#)
  - pack\_dms\_SLQSDmsSwiGetUimSelection, [1222](#)
  - pack\_dms\_SLQSDmsSwiIndicationRegister, [1222](#)
  - pack\_dms\_SLQSGetBandCapability, [1223](#)
  - pack\_dms\_SLQSGetERIFile, [1223](#)
  - pack\_dms\_SLQSGetPowerSaveModeConfig, [1223](#)
  - pack\_dms\_SLQSSetPowerSaveModeConfig, [1224](#)
  - pack\_dms\_SLQSSwiClearDyingGaspStatistics, [1224](#)
  - pack\_dms\_SLQSSwiGetCrashInfo, [1225](#)
  - pack\_dms\_SLQSSwiGetDyingGaspCfg, [1225](#)
  - pack\_dms\_SLQSSwiGetDyingGaspStatistics, [1225](#)
  - pack\_dms\_SLQSSwiGetFirmwareCurr, [1226](#)
  - pack\_dms\_SLQSSwiGetFwUpdateStatus, [1226](#)
  - pack\_dms\_SLQSSwiGetHostDevInfo, [1226](#)
  - pack\_dms\_SLQSSwiGetOSInfo, [1227](#)
  - pack\_dms\_SLQSSwiGetSerialNoExt, [1227](#)
  - pack\_dms\_SLQSSwiSetDyingGaspCfg, [1227](#)
  - pack\_dms\_SLQSSwiSetHostDevInfo, [1228](#)
  - pack\_dms\_SLQSSwiSetOSInfo, [1228](#)
  - pack\_dms\_SLQSUIMGetState, [1228](#)
  - pack\_dms\_SetActivationStatusCallback, [1218](#)
  - pack\_dms\_SetCrashAction, [1218](#)
  - pack\_dms\_SetCustFeature, [1219](#)
  - pack\_dms\_SetCustFeaturesV2, [1219](#)
  - pack\_dms\_SetEventReport, [1219](#)
  - pack\_dms\_SetFirmwarePreference, [1220](#)
  - pack\_dms\_SetIndicationRegister, [1220](#)
  - pack\_dms\_SetPower, [1220](#)
  - pack\_dms\_SetUSBComp, [1221](#)
  - pack\_dms\_SwiSetEventReport, [1229](#)
  - pack\_dms\_SwiUimSelect, [1229](#)
  - pack\_dms\_UIMChangePIN, [1229](#)
  - pack\_dms\_UIMGetControlKeyStatus, [1230](#)
  - pack\_dms\_UIMGetICCID, [1230](#)
  - pack\_dms\_UIMGetPINStatus, [1231](#)
  - pack\_dms\_UIMSetControlKeyProtection, [1231](#)
  - pack\_dms\_UIMSetPINProtection, [1231](#)
  - pack\_dms\_UIMUnblockControlKey, [1232](#)
  - pack\_dms\_UIMUnblockPIN, [1232](#)
  - pack\_dms\_UIMVerifyPIN, [1232](#)
  - pack\_dms\_ValidateSPC, [1233](#)
  - SPC\_SIZE, [1208](#)

- UNIQUE\_ID\_LEN, 1208
- unpack\_dms\_ActivateAutomatic, 1233
- unpack\_dms\_GetActivationState, 1234
- unpack\_dms\_GetBandCapability, 1234
- unpack\_dms\_GetCrashAction, 1234
- unpack\_dms\_GetCustFeature, 1235
- unpack\_dms\_GetCustFeaturesV2, 1235
- unpack\_dms\_GetDeviceCap, 1235
- unpack\_dms\_GetDeviceCapabilities, 1236
- unpack\_dms\_GetDeviceCapabilitiesV2, 1236
- unpack\_dms\_GetDeviceHardwareRev, 1236
- unpack\_dms\_GetDeviceMfr, 1237
- unpack\_dms\_GetDeviceSerialNumbers, 1237
- unpack\_dms\_GetFSN, 1238
- unpack\_dms\_GetFirmwareInfo, 1237
- unpack\_dms\_GetFirmwareRevision, 1238
- unpack\_dms\_GetFirmwareRevisions, 1238
- unpack\_dms\_GetHardwareRevision, 1239
- unpack\_dms\_GetIMSI, 1239
- unpack\_dms\_GetManufacturer, 1239
- unpack\_dms\_GetModelID, 1240
- unpack\_dms\_GetNetworkTime, 1240
- unpack\_dms\_GetNetworkTimeV2, 1240
- unpack\_dms\_GetOfflineReason, 1241
- unpack\_dms\_GetPRLVersion, 1241
- unpack\_dms\_GetPower, 1241
- unpack\_dms\_GetSerialNumbers, 1242
- unpack\_dms\_GetUSBComp, 1242
- unpack\_dms\_GetVoiceNumber, 1242
- unpack\_dms\_PSMCfgChange\_ind, 1243
- unpack\_dms\_ResetToFactoryDefaults, 1243
- unpack\_dms\_SLQSDmsSwiGetPCInfo, 1247
- unpack\_dms\_SLQSDmsSwiGetResetInfo, 1247
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind, 1248
- unpack\_dms\_SLQSDmsSwiGetUimSelection, 1248
- unpack\_dms\_SLQSDmsSwiIndicationRegister, 1248
- unpack\_dms\_SLQSGetBandCapability, 1249
- unpack\_dms\_SLQSGetBandCapabilityExt, 1249
- unpack\_dms\_SLQSGetERIFile, 1250
- unpack\_dms\_SLQSGetPowerSaveModeConfig, 1250
- unpack\_dms\_SLQSSetPowerSaveModeConfig, 1250
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics, 1251
- unpack\_dms\_SLQSSwiGetCrashInfo, 1251
- unpack\_dms\_SLQSSwiGetDyingGaspCfg, 1251
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics, 1252
- unpack\_dms\_SLQSSwiGetFirmwareCurr, 1252
- unpack\_dms\_SLQSSwiGetFwUpdateStatus, 1252
- unpack\_dms\_SLQSSwiGetHostDevInfo, 1253
- unpack\_dms\_SLQSSwiGetOSInfo, 1253
- unpack\_dms\_SLQSSwiGetSerialNoExt, 1253
- unpack\_dms\_SLQSSwiSetDyingGaspCfg, 1254
- unpack\_dms\_SLQSSwiSetHostDevInfo, 1254
- unpack\_dms\_SLQSSwiSetOSInfo, 1254
- unpack\_dms\_SLQSUIMGetState, 1255
- unpack\_dms\_SetActivationStatusCallback, 1244
- unpack\_dms\_SetCrashAction, 1244
- unpack\_dms\_SetCustFeature, 1244
- unpack\_dms\_SetCustFeaturesV2, 1245
- unpack\_dms\_SetEventReport, 1245
- unpack\_dms\_SetEventReport\_ind, 1245
- unpack\_dms\_SetFirmwarePreference, 1246
- unpack\_dms\_SetIndicationRegister, 1246
- unpack\_dms\_SetPower, 1246
- unpack\_dms\_SetUSBComp, 1247
- unpack\_dms\_SwiEventReportCallBack\_ind, 1255
- unpack\_dms\_SwiSetEventReport, 1255
- unpack\_dms\_SwiUimSelect, 1256
- unpack\_dms\_UIMChangePIN, 1256
- unpack\_dms\_UIMGetControlKeyStatus, 1256
- unpack\_dms\_UIMGetICCID, 1257
- unpack\_dms\_UIMGetPINStatus, 1257
- unpack\_dms\_UIMSetControlKeyProtection, 1257
- unpack\_dms\_UIMSetPINProtection, 1258
- unpack\_dms\_UIMUnblockControlKey, 1258
- unpack\_dms\_UIMUnblockPIN, 1258
- unpack\_dms\_UIMVerifyPIN, 1259
- unpack\_dms\_ValidateSPC, 1259
- dms\_ActivationStatusTlv, 56
  - activationStatus, 56
  - TlvPresent, 56
- dms\_LteBandsSupport, 64
  - lteBands, 65
  - supportedLteBandLen, 65
  - TLVPresent, 65
- dms\_OperatingModeTlv, 65
  - operatingMode, 66
  - TlvPresent, 66
- dms\_PSMActiveTimerIndTlv, 66
  - ActiveTimerInd, 66
  - TlvPresent, 66
- dms\_PSMActiveTimerTlv, 66
  - activeTimer, 67
  - TlvPresent, 67
- dms\_PSMDurationDueToOOSTlv, 67
  - durationDueToOOS, 67
  - TlvPresent, 67
- dms\_PSMDurationThresholdTlv, 67
  - durationThreshold, 68
  - TlvPresent, 68
- dms\_PSMEarlyWakeupTimeTlv, 68
  - earlyWakeupTime, 68
  - TlvPresent, 68
- dms\_PSMEnableStateIndTlv, 69
  - EnableStateInd, 69
  - TlvPresent, 69
- dms\_PSMEnableStateTlv, 69
  - enableState, 69
  - TlvPresent, 70
- dms\_PSMPeriodicUpdateTimerIndTlv, 70

- PeriodicUpdateTimerInd, [70](#)
- TlvPresent, [70](#)
- dms\_PSMPeriodicUpdateTimerTlv, [70](#)
  - periodicUpdateTimer, [71](#)
  - TlvPresent, [71](#)
- dms\_PSMRandomizationWindowTlv, [71](#)
  - randomizationWindow, [71](#)
  - TlvPresent, [71](#)
- dms\_TemperatureTlv, [71](#)
  - TempStat, [72](#)
  - Temperature, [72](#)
  - TlvPresent, [72](#)
- dms\_UimAutoSwitchActSlotTlv, [72](#)
  - TlvPresent, [72](#)
  - uimAutoSwitchActSlot, [72](#)
- dms\_UimStatusTlv, [73](#)
  - event, [73](#)
  - intf, [73](#)
  - TlvPresent, [73](#)
- dms\_VoltageTlv, [73](#)
  - TlvPresent, [74](#)
  - VoltStat, [74](#)
  - Voltage, [74](#)
- dms\_devCaps, [56](#)
  - DataServiceCapability, [57](#)
  - MaxRXChannelRate, [57](#)
  - MaxTXChannelRate, [57](#)
  - Radiolfaces, [57](#)
  - RadiolfacesSize, [57](#)
  - SimCapability, [57](#)
- dms\_devCurSubsCaps, [58](#)
  - CurSubsCapsLen, [58](#)
  - SubsCapList, [58](#)
- dms\_devMaxCfgListCaps, [58](#)
  - CurlIndex, [59](#)
  - DevCfgListLen, [59](#)
  - MaxActive, [59](#)
  - MaxSubs, [59](#)
  - SubsDevList, [59](#)
- dms\_devMaxSubsCaps, [59](#)
  - MaxSubsCapLen, [60](#)
  - MaxSubsList, [60](#)
- dms\_devMultiSimCaps, [60](#)
  - MaxSubs, [60](#)
  - SubsCfgList, [60](#)
  - SubsCfgListLen, [60](#)
- dms\_devMultiSimVoiceDataCaps, [61](#)
  - MaxActive, [61](#)
  - MaxSubs, [61](#)
- dms\_devSubsCfgList, [61](#)
  - MaxActive, [62](#)
  - SubsList, [62](#)
  - SubsListLen, [62](#)
- dms\_devSubsFeatureModeCaps, [62](#)
  - SubsFeatureLen, [63](#)
  - SubsFeatureList, [63](#)
- dms\_devSubsList, [63](#)
  - SubsList, [63](#)
  - SubsListLen, [63](#)
- dms\_devSubsVoiceDataCaps, [63](#)
  - SubsVoiceDataCapLen, [64](#)
  - SubsVoiceDataList, [64](#)
- dms\_devSubsVoiceDataList, [64](#)
  - SimVoiceDataCap, [64](#)
  - SubsVoiceDataCap, [64](#)
- domain
  - wds\_DomainNameList, [1168](#)
- domainLen
  - wds\_Domain, [1167](#)
- DomainList
  - unpack\_wds\_SLQSSetRuntimeSettings\_t, [1083](#)
- domainName
  - wds\_Domain, [1167](#)
- DormancyStat
  - wds\_DormStatTlv, [1168](#)
- dormancyStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- dormancyState
  - unpack\_wds\_GetDormancyState\_t, [1059](#)
- dormancyStatus
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [575](#)
  - unpack\_wds\_SLQSSetDUNCallInfo\_t, [1079](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- downLink
  - voice\_NSSAudioCtrl, [1146](#)
- dtmSupp
  - nas\_GSMSysInfo, [213](#)
- dtmSuppValid
  - nas\_GSMSysInfo, [213](#)
- dunchannelRate, [77](#)
  - CurrChanRxRate, [77](#)
  - CurrChanTxRate, [77](#)
  - MaxChanRxRate, [77](#)
  - MaxChanTxRate, [77](#)
- Duration
  - pack\_nas\_SetNetworkPreference\_t, [430](#)
  - unpack\_nas\_GetNetworkPreference\_t, [801](#)
- durationDueToOOS
  - dms\_PSMDurationDueToOOSTlv, [67](#)
- durationThreshold
  - dms\_PSMDurationThresholdTlv, [68](#)
- eAUDIO
  - common.h, [1196](#)
- eCAT
  - common.h, [1195](#)
- eCTL
  - common.h, [1195](#)
- eDMS
  - common.h, [1195](#)
- eFILE\_TYPE\_CAR\_PRI
  - lite-fw.h, [1287](#)
- eFILE\_TYPE\_COMPO\_PRI
  - lite-fw.h, [1287](#)
- eFILE\_TYPE\_NONE

- lite-fw.h, [1287](#)
- eFILE\_TYPE\_OEM\_PRI
  - lite-fw.h, [1287](#)
- eFIREHOSE\_ERR\_SECBOOT\_INVALID\_CERT\_CHA-  
IN
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_CWE
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_CWE\_NVU
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_INVALID
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_MBN
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_MBN\_GOBI
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_NVU
  - lite-fw.h, [1287](#)
- eFW\_TYPE\_SPK
  - lite-fw.h, [1287](#)
- eIMAGE\_TYPE\_ANY
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_APPL
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_APPS
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_BOOT
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_FILE
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_INVALID
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_KEYS
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_MAX
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_MIN
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_MODM
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_SPKG
  - lite-fw.h, [1288](#)
- eIMAGE\_TYPE\_USER
  - lite-fw.h, [1288](#)
- eIMS
  - common.h, [1195](#)
- eIMSA
  - common.h, [1195](#)
- eIND
  - common.h, [1196](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_1
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_10
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_11
  - nas.h, [1333](#)
- nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_12
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_125
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_126
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_127
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_13
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_14
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_17
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_18
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_19
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_2
  - nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_20
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_21
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_23
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_24
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_25
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_250
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_26
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_27
  - nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATIN-  
G\_BAND\_28
  - nas.h, [1334](#)

- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_29  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_3  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_30  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_31  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_32  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_33  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_34  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_35  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_36  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_37  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_38  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_39  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_4  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_40  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_41  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_42  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_43  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_46  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_47  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_48  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_5  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_6  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_66  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_7  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_71  
nas.h, [1334](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_8  
nas.h, [1333](#)
- eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPERATING\_BAND\_9  
nas.h, [1333](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED  
nas.h, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED  
nas.h, [1334](#)
- eLOC  
common.h, [1195](#)
- eLOG\_DEBUG  
common.h, [1195](#)
- eLOG\_FATAL  
common.h, [1195](#)
- eLOG\_INFO  
common.h, [1195](#)
- eLOG\_WARN  
common.h, [1195](#)
- eModel\_9X15  
lite-fw.h, [1288](#)
- eModel\_9X30

- lite-fw.h, [1288](#)
- eModel\_9x06
  - lite-fw.h, [1288](#)
- eModel\_9x07
  - lite-fw.h, [1288](#)
- eModel\_Unknown
  - lite-fw.h, [1288](#)
- eModel\_WP9X15
  - lite-fw.h, [1288](#)
- eNAS
  - common.h, [1195](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_ACTIVATED\_LITE
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGURED\_DEACTIVATED\_LITE
  - nas.h, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIGURED\_LITE
  - nas.h, [1335](#)
- ePDS
  - common.h, [1195](#)
- eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_BUFFER\_SZ
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_CANCEL\_OP
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_DRIVER
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_ENUM\_BEGIN
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_ENUM\_END
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_FILE\_COPY
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_FILE\_OPEN
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_GENERAL
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INTERNAL
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INVALID\_ARG
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INVALID\_DEVID
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INVALID\_FILE
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INVALID\_QMI\_RSP
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_INVALID\_XID
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_MEMORY
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_MULTIPLE\_DEVICES
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_NO\_CANCELABLE\_OP
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_NO\_CONNECTION
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_NO\_DEVICE
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_NO\_SIGNAL
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_NONE
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_NULL\_TLV
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_OFFLINE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_PDU\_GENERATION
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_ABORTED
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_CALL\_FAILED
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_CAT\_END
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_CAT\_START
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_CAUSE\_CODE
  - qmerrno.h, [1453](#)



- eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_CONNECT
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_DISABLED
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_ENCODING
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_GENERAL
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_IFACE
  - qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INTERNAL
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ARG
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_ID
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_INDEX
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTION
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP
  - qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_MAX
  - qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_USE
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAILURE
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT
  - qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG
  - qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED
  - qmerrno.h, [1454](#)

- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_OFFSET  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_REQ  
qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH  
qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO  
qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUPPORTED  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_RSP  
qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO  
qmerrno.h, [1451](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN  
qmerrno.h, [1453](#)
- eQCWWAN\_ERR\_QMI\_WIDTH  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_RESET  
qmerrno.h, [1452](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_END  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS  
qmerrno.h, [1454](#)



- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_START  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP  
qmerrno.h, [1454](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_END  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_START  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_END  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOAD\_MODE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_INVALID\_SLOT\_INDEX  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT\_ACTIVE\_IMAGE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_IMAGE\_NOT\_SIGNED  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_INVALID\_CERT\_CHAIN  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_TOO\_MANY\_FILES  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_START  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISM\_END  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED  
qmerrno.h, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_START  
qmerrno.h, [1455](#)
- eQDL\_HW\_FLOW\_DISABLE  
lite-fw.h, [1288](#)
- eQDL\_HW\_FLOW\_ENABLE  
lite-fw.h, [1288](#)
- eQDL\_HW\_FLOW\_INIT  
lite-fw.h, [1288](#)
- eQDL\_HW\_FLOW\_Unknown  
lite-fw.h, [1288](#)
- eQDL\_MODE\_INIT  
lite-fw.h, [1288](#)
- eQDL\_MODE\_TTYUSB  
lite-fw.h, [1288](#)
- eQDL\_MODE\_UART0  
lite-fw.h, [1288](#)
- eQDL\_MODE\_Unknown  
lite-fw.h, [1288](#)
- eQMI\_LOC\_SESS\_STATUS\_FAILURE  
loc.h, [1301](#)
- eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS  
loc.h, [1301](#)
- eQMI\_LOC\_SESS\_STATUS\_SUCCESS  
loc.h, [1301](#)
- eQMI\_LOC\_SESS\_STATUS\_TIMEOUT  
loc.h, [1301](#)
- eQOS  
common.h, [1195](#)
- eREQ  
common.h, [1196](#)

- eRMS
  - common.h, [1195](#)
- eRSP
  - common.h, [1196](#)
- eSAR
  - common.h, [1195](#)
- eSDP\_FWDWL\_ERR\_END
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_FAIL
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_FW\_UPGRADE
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_FW\_VERSION\_FAIL
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_GENERAL
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_INVALID\_DEV
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_INVALID\_PATH
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_PATH\_NOT\_SPECIFIED
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_PATH\_TOO\_LONG
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_PRI\_FAIL
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_SDK
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_SDP\_TIMEOUT
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_SET\_CBK
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_ERR\_TIMEOUT
  - lite-fw.h, [1287](#)
- eSDP\_FWDWL\_SUCCESS
  - lite-fw.h, [1287](#)
- eSMS
  - common.h, [1195](#)
- eSWIAUDIO
  - common.h, [1196](#)
- eSWIDMS
  - common.h, [1196](#)
- eSWILOC
  - common.h, [1196](#)
- eSWIM2MCMD
  - common.h, [1196](#)
- eSWIM2MCMD\_AVC2
  - common.h, [1196](#)
- eSWIOMA
  - common.h, [1195](#)
- eSWIOMAEEXT
  - common.h, [1196](#)
- eTIMEOUT\_10\_S
  - common.h, [1196](#)
- eTIMEOUT\_20\_S
  - common.h, [1196](#)
- eTIMEOUT\_2\_S
  - common.h, [1196](#)
- eTIMEOUT\_300\_S
  - common.h, [1196](#)
- eTIMEOUT\_30\_S
  - common.h, [1196](#)
- eTIMEOUT\_5\_S
  - common.h, [1196](#)
- eTIMEOUT\_60\_S
  - common.h, [1196](#)
- eTIMEOUT\_8\_S
  - common.h, [1196](#)
- eTIMEOUT\_DEFAULT
  - common.h, [1196](#)
- eTMD
  - common.h, [1195](#)
- eUIM
  - common.h, [1195](#)
- eVOICE
  - common.h, [1195](#)
- eWDS
  - common.h, [1195](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_END
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM

- qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_LEN\_INVALID
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NOT\_INITED
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL
  - qmerrno.h, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END
  - qmerrno.h, [1456](#)
- EAWTlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- ECIOThresListLen
  - nas\_ECIOThresh, [199](#)
- ECTCallState
  - voice\_ECTNum, [1141](#)
- eLOG\_LEVEL
  - common.h, [1195](#)
- EMTlv
  - NASQmiCbkNasSystemSelPrefInd, [368](#)
- eQCWWANError
  - qmerrno.h, [1451](#)
- eQMI\_SVC
  - common.h, [1195](#)
- ERI\_DATA\_MAX\_SIZE
  - dms.h, [1207](#)
- ESATlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- ESAWTlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- ESNString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [677](#)
- ETWSPLMNInfo
  - eTWSPLMNInfoTlv, [78](#)
- eTWSPLMNInfoTlv, [78](#)
  - ETWSPLMNInfo, [78](#)
  - TlvPresent, [78](#)
- ETWSPLMNTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [935](#)
- ETWSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [935](#)
- eTimeout
  - common.h, [1196](#)
- eValid
  - LibPackTFTIDParams, [143](#)
- EarMute
  - pack\_audio\_SLQSSetAudioProfile\_t, [377](#)
  - pack\_swiaudio\_SLQSSetM2MAVMute\_t, [503](#)
  - unpack\_audio\_SLQSGetAudioProfile\_t, [658](#)
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [951](#)
  - unpack\_swiaudio\_SLQSGetM2MAVMute\_t, [953](#)
- earfcn
  - nas\_infoInterFreq, [222](#)
  - nas\_LTEInfoIntraFreq, [234](#)
  - nas\_ltePCI, [243](#)
  - nas\_umtsLTENbrCell, [342](#)
- earfcn0
  - nas\_lteEARFCN, [227](#)
- earfcn1
  - nas\_lteEARFCN, [227](#)
- earlyWakeupTime
  - dms\_PSMEarlyWakeupTimeTlv, [68](#)
- ecio
  - cdmaSSInfo, [53](#)
  - hdrSSInfo, [83](#)
  - nas\_ecioListElement, [198](#)
  - nas\_rxInfo, [314](#)
  - nas\_UMTSExtInfo, [337](#)
  - nas\_UMTSInfo, [339](#)
  - tdscdmaSigInfoExt, [619](#)
- ecioDelta
  - nas\_SLQSSignalStrengthsIndReq, [322](#)
- ecioInfo
  - nas\_SLQSSignalStrengthsInformation, [323](#)
- ecioList
  - unpack\_nas\_SLQSGetSignalStrength\_t, [833](#)
- ecioListLen
  - unpack\_nas\_SLQSGetSignalStrength\_t, [833](#)
- ecioThresholdList
  - nas\_SLQSSignalStrengthsIndReq, [322](#)
- ecioThresholdListLen
  - nas\_SLQSSignalStrengthsIndReq, [322](#)
- edrxCiotLteModeTlv
  - unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t, [854](#)
- edrxCycleLengthTlv
  - unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t, [854](#)
- edrxEnableTypeTlv
  - unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t, [854](#)
- edrxEnabled
  - nas\_EdrxEnableType, [201](#)
- edrxPagingTimeWindowTlv
  - unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t, [855](#)
- edrxPtw
  - nas\_EdrxPagingTimeWindow, [201](#)
- edrxRatType
  - nas\_EdrxRatType, [202](#)
- edrxRatTypeTlv
  - unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t, [855](#)
- egprsSupp
  - nas\_GSMSSysInfo, [213](#)
- egprsSuppValid
  - nas\_GSMSSysInfo, [213](#)
- ehrpMTUSize

- swidms\_ehrpdMTUSizeTlv, 614
- elevation
  - loc\_satelliteInfo, 165
- EmerMode
  - nas\_EmerModeTlv, 203
  - NASEmergencyModeTlv, 354
- emmConnState
  - nas\_LTEInfo, 232
- emmState
  - nas\_LTEInfo, 232
- emmSubState
  - nas\_LTEInfo, 232
- Enable
  - pack\_swiaudio\_SLQSSetM2MAudioLPBK\_t, 500
- enable
  - pack\_pds\_SetPDSSState\_t, 476
  - pack\_qos\_SLQSSetQosEventCallback\_t, 487
  - pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, 509
  - swidms\_SwiDmsGetHWWatchdog, 618
- EnableState
  - unpack\_dms\_PSMCfgChange\_ind\_t, 694
- enableState
  - dms\_PSMEnableStateTlv, 69
- EnableStateInd
  - dms\_PSMEnableStateIndTlv, 69
- enabled
  - pack\_rms\_SetSMSWake\_t, 487
  - unpack\_rms\_GetSMSWake\_t, 929
  - unpack\_wds\_GetMobileIPProfile\_t, 1062
- EnabledStatus
  - unpack\_pds\_SetPdsState\_Ind\_t, 900
- EncryptProt
  - nas\_protocolSubtypeElement, 290
- EncryptedPIN1
  - pack\_uim\_ChangePin\_t, 521
  - pack\_uim\_SetPinProtection\_t, 523
  - pack\_uim\_UnblockPin\_t, 534
- EndPointType
  - qos\_BindDataPortPeripheralEndPointID\_t, 589
- EndProactiveSession
  - cat\_EndProactiveSessionTlv, 51
- EngineState
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- engineState
  - unpack\_loc\_EngineState\_Ind\_t, 770
- eqmiCbKSetStatus
  - sms.h, 1474
- eriData
  - eriDataparams, 78
- eriDataLen
  - eriDataparams, 78
- eriDataparams, 77
  - eriData, 78
  - eriDataLen, 78
- eriFile
  - unpack\_dms\_SLQSGetERIFile\_t, 716
- error
  - unpack\_wds\_GetLastMobileIPError\_t, 1059
- errorClass
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t, 949
- ErrorCode
  - imsa\_IMSRegStatusErrorCodeInfo, 98
- ErrorCodeData
  - imsa\_RatHandoverStatusInfo, 100
- ErrorCodeLen
  - imsa\_RatHandoverStatusInfo, 100
- errorMask
  - unpack\_cat\_SetCATEventCallback\_t, 661
- errorRate
  - nas\_errorRateListElement, 204
- errorRateInfo
  - nas\_SLQSSignalStrengthsInformation, 323
- errorRateList
  - unpack\_nas\_SLQSGetSignalStrength\_t, 833
- errorRateListLen
  - unpack\_nas\_SLQSGetSignalStrength\_t, 833
- errorState
  - slotInf, 594
  - uim\_slotInfo, 649
- esn
  - unpack\_dms\_GetSerialNumbers\_t, 690
- esnSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 676
- EspSpi
  - unpack\_qos\_swiQosFilter\_t, 921
- EtwsMessageInfo
  - sMSEtwsMessageTlv, 610
- event
  - dms\_UimStatusTlv, 73
  - unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, 917
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 919
- event\_Index
  - unpack\_cat\_SetCatEventCallback\_ind\_t, 660
- EventID
  - cat\_commonEventTlv, 50
- EventLength
  - cat\_commonEventTlv, 50
- eventMask
  - pack\_cat\_SetCATEventCallback\_t, 380
  - pack\_uim\_SLQSUIEventRegister\_t, 525
  - unpack\_uim\_SLQSUIEventRegister\_t, 991
- eventRegister
  - pack\_loc\_EventRegister\_t, 411
- eventType
  - unpack\_swima\_SLQSOMADMAAlertCallback\_ind\_t, 972
- evrcCapability
  - voice\_prefVoiceSO, 1150
- executingImage
  - FMSImageIDEntries, 81
- exponent
  - unpack\_qos\_pktErrRate\_t, 909

- extBit
  - voice\_calledPartySubAdd, [1123](#)
- extDispInfo
  - voice\_extDispRecInfo, [1141](#)
- extDispInfoLen
  - voice\_extDispRecInfo, [1141](#)
- ExtErrorCode
  - PackCreateProfileOut, [581](#)
- extPowerState
  - pack\_loc\_SetExtPowerState\_t, [411](#)
- extendedErrorCode
  - unpack\_wds\_SLQSDDeleteProfile\_t, [1070](#)
- FDNStatus
  - uim\_UIMGetFDNStatus, [650](#)
- FIRMWARE\_BCHDATESIZE
  - lite-fw.h, [1284](#)
- FLOAT
  - SwiDataTypes.h, [1517](#)
- FMSImageElement, [78](#)
  - buildId, [79](#)
  - buildIdLength, [79](#)
  - imageId, [79](#)
  - imageType, [79](#)
- FMSImageIDEntries, [80](#)
  - executingImage, [81](#)
  - imageIDElement, [81](#)
  - imageIDSize, [81](#)
  - imageType, [81](#)
  - maxImages, [81](#)
- FMSImageIDElement, [79](#)
  - buildID, [80](#)
  - buildIDLength, [80](#)
  - failureCount, [80](#)
  - imageID, [80](#)
  - storageIndex, [80](#)
- FMSImageList, [81](#)
  - imageIDEntries, [81](#)
  - listSize, [81](#)
- FMSPrefImageList, [82](#)
  - listEntries, [82](#)
  - listSize, [82](#)
- FOTAUpdate
  - pack\_swioima\_SLQSOMADMSetSettings\_t, [517](#)
  - pack\_swioima\_SLQSOMADMSetSettingsExt\_t, [518](#)
  - unpack\_swioima\_SLQSOMADMGetSettings\_t, [980](#)
- FOTAdownload
  - pack\_swioima\_SLQSOMADMSetSettings\_t, [517](#)
  - pack\_swioima\_SLQSOMADMSetSettingsExt\_t, [518](#)
  - unpack\_swioima\_SLQSOMADMGetSettings\_t, [980](#)
- FUMOEnable
  - pack\_swioima\_SLQSOMADMSetSettingsExt\_t, [518](#)
- facility
  - pack\_dms\_UIMGetControlKeyStatus\_t, [395](#)
  - pack\_dms\_UIMSetControlKeyProtection\_t, [396](#)
  - pack\_dms\_UIMUnblockControlKey\_t, [397](#)
- facilityCk
  - pack\_dms\_UIMSetControlKeyProtection\_t, [396](#)
  - pack\_dms\_UIMUnblockControlKey\_t, [397](#)
- facilityState
  - pack\_dms\_UIMSetControlKeyProtection\_t, [396](#)
  - unpack\_dms\_UIMGetControlKeyStatus\_t, [731](#)
- FailErrCode
  - unpack\_imsa\_SLQSImsaPdpStatusCallback\_ind\_t, [757](#)
- failureCount
  - FMSImageIDElement, [80](#)
- family
  - pack\_wds\_GetDefaultProfileNum\_t, [556](#)
  - pack\_wds\_SetDefaultProfileNum\_t, [563](#)
- feature
  - uim\_depersonalizationInformation, [631](#)
  - uim\_personalizationStatus, [638](#)
- fileID
  - uim\_fileAttributes, [634](#)
  - uim\_fileInfo, [635](#)
- fileIndex
  - pack\_uim\_ReadTransparent\_t, [522](#)
  - pack\_uim\_SLQSUIMGetFileAttributes\_t, [526](#)
  - pack\_uim\_SLQSUIMReadRecord\_t, [529](#)
  - pack\_uim\_SLQSUIMWriteRecord\_t, [533](#)
  - pack\_uim\_SLQSUIMWriteTransparent\_t, [534](#)
- fileSize
  - uim\_fileAttributes, [634](#)
- fileType
  - uim\_fileAttributes, [634](#)
- fill\_pack\_ctx
  - common.h, [1196](#)
- fill\_sdu\_hdr
  - common.h, [1196](#)
- filterId
  - LibPackTFTIDParams, [143](#)
- fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [513](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [971](#)
- fix\_rate\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [971](#)
- fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, [513](#)
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [971](#)
- fix\_type\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, [971](#)
- fixAccuracy
  - pack\_pds\_StartPDSTrackingSessionExt\_t, [484](#)
- fixCount
  - pack\_pds\_StartPDSTrackingSessionExt\_t, [484](#)
- fixInterval
  - pack\_pds\_StartPDSTrackingSessionExt\_t, [484](#)
- fixTimeout
  - pack\_pds\_StartPDSTrackingSessionExt\_t, [484](#)
- flags
  - sensorData\_t, [591](#)
- flowLabel
  - LibPackTFTIDParams, [143](#)

- fms.h, 1260
  - GetValidFwPriCombinations, 1261
  - pack\_fms\_GetImagesPreference, 1261
  - pack\_fms\_GetStoredImages, 1262
  - pack\_fms\_SetImagesPreference, 1262
  - unpack\_fms\_GetImagesPreference, 1262
  - unpack\_fms\_GetStoredImages, 1262
  - unpack\_fms\_SetImagesPreference, 1263
- Forbidden
  - nas\_QmiNas3GppNetworkInfo, 292
- forbiddenNwlnstLen
  - nas\_ForbiddenNetworks3GPP, 205
- forceFlag
  - pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, 479
- ForceRev0
  - unpack\_nas\_GetCDMANetworkParameters\_t, 796
- format
  - sMSTransferRouteMTMessageInfo, 613
- ForwardMac
  - nas\_protocolSubtypeElement, 290
- fqdnAddr
  - wds\_PCSCFFQDNAddress, 1172
- fqdnLen
  - wds\_PCSCFFQDNAddress, 1172
- freeSlots
  - sms\_maxStorageSizeResp, 600
- freq
  - nas\_PhyCaAggPcellInfo, 280
  - nas\_PhyCaAggScellIndType, 282
  - nas\_PhyCaAggScellInfo, 284
  - nas\_QmisNasSlqsNasPCICellInfo, 294
  - NASPhyCaAggPcellInfo, 361
  - NASPhyCaAggScellArray, 362
  - NASPhyCaAggScellIndType, 364
  - NASPhyCaAggScellInfo, 365
- freqsLen
  - nas\_LTEInfoInterfreq, 232
  - nas\_LTEInfoNeighboringGSM, 235
  - nas\_LTEInfoNeighboringWCDMA, 236
- fromServiceId
  - sms\_BroadcastConfig, 595
- fullPath
  - \_litelw\_FirmwareFileInfo, 36
- fumoResultCode
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, 978
- fumoState
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, 978
- function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- function\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- FwAutoCheck
  - unpack\_swima\_SLQSOMADMGetSettings\_t, 980
- FwAutoSDM
  - unpack\_swivms\_SLQSAVMSGetSettings\_t, 957
  - unpack\_swivms\_SLQSAVMSGetSettings\_v2\_t, 960
- FwAvailability
  - unpack\_swima\_SLQSOMADMStartSession\_t, 981
- FwPromptUpdate
  - unpack\_swivms\_SLQSAVMSGetSettings\_t, 958
  - unpack\_swivms\_SLQSAVMSGetSettings\_v2\_t, 960
- FwPromptdownload
  - unpack\_swivms\_SLQSAVMSGetSettings\_t, 957
  - unpack\_swivms\_SLQSAVMSGetSettings\_v2\_t, 960
- fwloadsize
  - unpack\_omaDmFotaTlv\_t, 891
- fwloadComplete
  - unpack\_omaDmFotaTlv\_t, 891
- fwvers
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 721
- gDIBitRate
  - LibPackQosClassID, 142
- GPRSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1083
- GPSPM
  - pack\_dms\_SetCustFeature\_t, 385
  - unpack\_dms\_GetCustFeature\_t, 667
- GPSSel
  - pack\_dms\_SetCustFeature\_t, 385
  - unpack\_dms\_GetCustFeature\_t, 667
- GSMRSSIThreshListLen
  - nas\_GSMRSSIThresh, 210
- GSMSSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 865
- gUIBitRate
  - LibPackQosClassID, 142
- GWAOPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- GWAcqOrderPref
  - nas\_GWAcqOrderPrefTlv, 214
  - NASGWAcqOrderPrefTlv, 356
- GWAddressV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1083
- gcdumpString
  - crashInformation, 55
- gcdumpStrlen
  - crashInformation, 55
- Generator
  - pack\_audio\_SLQSGetAudioProfile\_t, 374
  - pack\_audio\_SLQSGetAudioVoITLBConfig\_t, 374
  - pack\_audio\_SLQSSetAudioProfile\_t, 377
  - pack\_audio\_SLQSSetAudioVoITLBConfig\_t, 378
  - pack\_swiaudio\_SLQSGetM2MAudioVolume\_t, 498
  - pack\_swiaudio\_SLQSSetM2MAudioVolume\_t, 502
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, 951
- geoSysIdx
  - nas\_AddCDMASysInfo, 173



- nas\_AddSysInfo, [174](#)
- geranArfcn
  - nas\_geranInstArr, [207](#)
  - nas\_geranInstInfo, [208](#)
- geranBsicBcc
  - nas\_geranInstArr, [207](#)
  - nas\_geranInstInfo, [208](#)
- geranBsicNcc
  - nas\_geranInstArr, [207](#)
  - nas\_geranInstInfo, [208](#)
- geranInst
  - nas\_UMTSExtInfo, [337](#)
  - nas\_UMTSInfo, [339](#)
- geranInstElement
  - nas\_UMTSExtInfo, [337](#)
- GeranInstInfo
  - nas\_UMTSInfo, [339](#)
- geranRank
  - nas\_geranInstArr, [207](#)
- geranRssi
  - nas\_geranInstArr, [207](#)
  - nas\_geranInstInfo, [208](#)
- get\_version
  - common.h, [1196](#)
- getAlIcAllInfo
  - voice\_arrCallInfo, [1116](#)
- GetAlIcAllRmtPtyName
  - voice\_arrRemotePartyName, [1118](#)
- GetCustomFeatureV2
  - unpack\_dms\_GetCustFeaturesV2\_t, [668](#)
- GetValidFwPriCombinations
  - fms.h, [1261](#)
- glo\_almanac\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- glo\_ephemeris\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- glo\_health\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- glo\_visible\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- GlobalCellID
  - nas\_QmisNasSlqsNasPCICellInfo, [294](#)
- globalCellId
  - nas\_LTEInfoIntrafreq, [234](#)
- glog
  - common.h, [1198](#)
- gloglvl
  - common.h, [1198](#)
- gnssSvId
  - loc\_satelliteInfo, [165](#)
- gnssSvUsedList
  - loc\_svUsedforFix, [168](#)
- gnssSvUsedList\_len
  - loc\_svUsedforFix, [168](#)
- Gpp2TimeZone
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- GppNetworkDSTAdjustment
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- GppTimeZone
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- gps\_almanac\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- gps\_ephemeris\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- gps\_health\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- gps\_visible\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- GpsEnable
  - pack\_dms\_SetCustFeature\_t, [385](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
- gpsTimeOfWeekMs
  - loc\_gpsTime, [156](#)
- gpsWeek
  - loc\_gpsTime, [156](#)
- grntDownlinkBitrate
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMInQoS, [1179](#)
- grntUplinkBitrate
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMInQoS, [1179](#)
- gsmAmrStat
  - voice\_curAMRConfig, [1138](#)
- GsmCellInfo
  - nas\_lteGsmCellInfo, [230](#)
- gsmUmtsDI
  - pack\_nas\_SLQSNasSwilIndicationRegister\_t, [454](#)
- gsmUmtsUI
  - pack\_nas\_SLQSNasSwilIndicationRegister\_t, [454](#)
- guaranteedRate
  - unpack\_qos\_dataRate\_t, [907](#)
- gwAddressV6
  - wds\_IPV6GWAddressInfo, [1171](#)
- gwV6PrefixLen
  - wds\_IPV6GWAddressInfo, [1171](#)
- gyroData
  - pack\_loc\_SLQSLOCInjectSensorData\_t, [419](#)
- gyroTemp
  - pack\_loc\_SLQSLOCInjectSensorData\_t, [419](#)
- gyroTimeSrc
  - pack\_loc\_SLQSLOCInjectSensorData\_t, [420](#)
- HASPI
  - unpack\_wds\_GetMobileIPProfile\_t, [1062](#)
- HASState
  - unpack\_wds\_GetMobileIPProfile\_t, [1062](#)
- HDOP
  - loc\_precisionDilution, [163](#)
- HDRECIOThreshListLen
  - nas\_HDRECIOThresh, [214](#)
- HDRIOTThreshListLen
  - nas\_HDRIOTThresh, [215](#)
- HDRRSSIThreshListLen
  - nas\_HDRRSSIThresh, [216](#)

HDRSINRThresListLen  
     nas\_HDRSINRThresh, 217  
 HDRSINRThreshListLen  
     nas\_HDRSINRThreshold, 217  
 HDRSSInfo  
     unpack\_nas\_SLQSNasGetSigInfo\_t, 865  
 HWVersion  
     nas\_DeviceConfigDetail, 197  
 HardwareControlledMode  
     unpack\_dms\_GetPower\_t, 688  
 has\_LpmFlag  
     unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 702  
 has\_PdpStatusConfig  
     pack\_imsa\_SLQSRegisterIMSAIndication\_t, 408  
 has\_PersistentLpm  
     unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 702  
 has\_PositionDataNMEA  
     unpack\_pds\_SetEventReport\_Ind\_t, 899  
 has\_PowerOffMode  
     unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 702  
 has\_RatHandoverStatusConfig  
     pack\_imsa\_SLQSRegisterIMSAIndication\_t, 408  
 has\_RegStatusConfig  
     pack\_imsa\_SLQSRegisterIMSAIndication\_t, 408  
 has\_ServiceStatusConfig  
     pack\_imsa\_SLQSRegisterIMSAIndication\_t, 408  
 has\_SessionStatus  
     unpack\_pds\_SetEventReport\_Ind\_t, 899  
 has\_Wdisable  
     unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 702  
 has\_accelTemp  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_acceleroTimeSrc  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_accleroData  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_altitudeSrcInfo  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_altitudeWrtEllipsoid  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_altitudeWrtMeanSeaLevel  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_confidence  
     pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 421  
 has\_dLatitude  
     unpack\_pds\_SetEventReport\_Ind\_t, 899  
 has\_dLongitude  
     unpack\_pds\_SetEventReport\_Ind\_t, 899  
 has\_gyroData  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_gyroTemp  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_gyroTimeSrc  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_horConfidence  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_horReliability  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_horUncCircular  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_latitude  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_longitude  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_opaqueId  
     pack\_loc\_SLQSLOCInjectSensorData\_t, 420  
 has\_posSrc  
     unpack\_pds\_SetEventReport\_Ind\_t, 899  
 has\_positionSrc  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_rawHorConfidence  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_rawHorUncCircular  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_serverAddrTypeMask  
     pack\_loc\_SLQSLOCGetServer\_t, 413  
 has\_timestampAge  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_timestampUtc  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_vertConfidence  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_vertRelicability  
     pack\_loc\_SLQSLOCInjectPosition\_t, 417  
 has\_vertUnc  
     pack\_loc\_SLQSLOCInjectPosition\_t, 418  
 hdrActiveProt  
     nas\_HDRSysInfo, 219  
 hdrActiveProtValid  
     nas\_HDRSysInfo, 219  
 hdrHybrid  
     nas\_detailSvcInfo, 196  
 HdrPersonality  
     unpack\_nas\_SLQSGetServingSystem\_t, 826  
     unpack\_nas\_SLQSGetServingSystemV2\_t, 830  
 hdrPersonality  
     nas\_HDRSysInfo, 219  
     NASServingSystemInfo, 371  
 hdrPersonalityValid  
     nas\_HDRSysInfo, 219  
 hdrSSInfo, 82  
     ecio, 83  
     io, 83  
     rssi, 83  
     sinr, 83  
 hdrSrvStatus  
     nas\_detailSvcInfo, 196  
 headerType  
     \_litefw\_FirmwareFileInfo, 36  
 healthStatus  
     loc\_satelliteInfo, 165  
 helper\_get\_error\_code  
     common.h, 1196  
 helper\_get\_error\_reason  
     common.h, 1197  
 helper\_get\_req\_str



- common.h, [1197](#)
- helper\_get\_resp\_ctx
  - common.h, [1197](#)
- helper\_get\_xid
  - common.h, [1197](#)
- helper\_isBootLoader\_DebugEnabled
  - common.h, [1197](#)
- helper\_set\_log\_func
  - common.h, [1197](#)
- helper\_set\_log\_lvl
  - common.h, [1198](#)
- hfaMaxRetry
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- hfaRetryIndex
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- hfaRetryInterval
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- hfaStatus
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- hiddenKey
  - uim\_UIMGetHiddenKeyStatus, [650](#)
- homeOrigVoiceSO
  - voice\_prefVoiceSO, [1150](#)
- homePageVoiceSO
  - voice\_prefVoiceSO, [1150](#)
- horConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- horReliability
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- horUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- HorizontalUncertainty
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- hostID
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, [392](#)
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, [724](#)
- hotSwap
  - uim\_hotSwapStatus, [637](#)
- hotSwapLength
  - uim\_hotSwapStatus, [637](#)
- hour
  - nas\_timeInfo, [334](#)
  - nas\_UniversalTime, [343](#)
- hrpdMTUSize
  - swidms\_hrpdmTUSizeTlv, [615](#)
- hsCallStatus
  - nas\_WCDMASysInfo, [351](#)
- hsCallStatusValid
  - nas\_WCDMASysInfo, [351](#)
- hsInd
  - nas\_WCDMASysInfo, [351](#)
- hsIndValid
  - nas\_WCDMASysInfo, [351](#)
- hwType
  - wds\_DHCPv4HWConfig, [1165](#)
  - wdsDhcpv4HwConfig, [1180](#)
- hwVer
  - unpack\_dms\_GetHardwareRevision\_t, [682](#)
- IDTlv
  - unpack\_ims\_SLQSUserCfgCallBack\_ind\_t, [751](#)
- iFaceTab
  - swiaudio\_PCMparams, [614](#)
- iFaceTabLen
  - swiaudio\_PCMparams, [614](#)
- iLTEbandValue
  - nas\_PhyCaAggPcellInfo, [280](#)
  - nas\_PhyCaAggScellInfo, [284](#)
  - NASPhyCaAggPcellInfo, [361](#)
  - NASPhyCaAggScellInfo, [365](#)
- IMCNflag
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)
- IMEIString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [677](#)
- IMG\_MASK\_CLEAR
  - lite-fw.h, [1284](#)
- IMG\_MASK\_GENERIC
  - lite-fw.h, [1284](#)
- IMG\_MASK\_MDM
  - lite-fw.h, [1284](#)
- IMG\_MASK\_PRI
  - lite-fw.h, [1284](#)
- IMSInfo
  - sMSOnIMSTlv, [612](#)
- IMSRegistration
  - unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t, [758](#)
- IMSRegistrationError
  - unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t, [758](#)
- IMSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [935](#)
- INT32
  - SwiDataTypes.h, [1517](#)
- INT8
  - SwiDataTypes.h, [1517](#)
- IOThresListLen
  - nas\_IOThresh, [223](#)
- IPAddress
  - nas\_DataStatusDetail, [194](#)
- IPAddressV6
  - ipv6AddressInfo, [104](#)
  - wds\_IPV6AddressInfo, [1170](#)
- IPFamSupport
  - pack\_dms\_SetCustFeature\_t, [385](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
- IPFamilyPreference
  - pack\_wds\_SLQSSetIPFamilyPreference\_t, [574](#)
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)
- IPSECSPI
  - LibPackTFTIDParams, [143](#)
- IPV6AddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)

- IPv6GWAddrInfo
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- IPv6PrefixLen
  - ipv6AddressInfo, 105
  - wds\_IPv6AddressInfo, 1170
- IPv4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- IPv4Addr
  - loc\_IPv4Config, 159
  - wds\_IPv4AdTlv, 1170
- IPv4AddrTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, 1049
- IPv4DstAddr
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv4Port
  - loc\_IPv4Config, 159
- IPv4SrcAddr
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv4Tos
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv6Addr
  - loc\_IPv6Config, 160
- IPv6DstAddr
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv6Label
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv6Port
  - loc\_IPv6Config, 160
- IPv6SrcAddr
  - unpack\_qos\_swiQosFilter\_t, 921
- IPv6TrafCls
  - unpack\_qos\_swiQosFilter\_t, 921
- Id
  - unpack\_qos\_swiQosFilter\_t, 921
- id
  - loc\_BdsSV, 150
  - loc\_SV, 167
  - nas\_CSGID, 191
  - pack\_dms\_UIMChangePIN\_t, 394
  - pack\_dms\_UIMSetPINProtection\_t, 397
  - pack\_dms\_UIMUnblockPIN\_t, 398
  - pack\_dms\_UIMVerifyPIN\_t, 398
  - unpack\_qos\_QosFlowInfoState\_t, 911
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 919
- id\_length
  - DMScustSettingInfo, 75
- IdleState
  - nas\_protocolSubtypeElement, 290
- IfaceID
  - qos\_BindDataPortPeripheralEndPointID\_t, 589
- image\_info\_t, 83
  - buildID, 84
  - buildIDLen, 84
  - imageType, 84
  - uniqueID, 84
- imageID
  - FMSImageIDElement, 80
- imageIDElement
  - FMSImageIDEntries, 81
- imageIDEntries
  - FMSImageList, 81
- imageIDSize
  - FMSImageIDEntries, 81
- imageId
  - FMSImageElement, 79
- imageList
  - unpack\_fms\_GetStoredImages\_t, 737
- ImageListSize
  - unpack\_fms\_GetImagesPreference\_t, 736
- imageListSize
  - pack\_fms\_SetImagesPreference\_t, 400
- imageMask
  - \_litefw\_FirmwareFileInfo, 36
- imageType
  - \_litefw\_FirmwareFileInfo, 36
  - FMSImageElement, 79
  - FMSImageIDEntries, 81
  - image\_info\_t, 84
- ImageTypes
  - unpack\_fms\_SetImagesPreference\_t, 737
- ImageTypesSize
  - unpack\_fms\_SetImagesPreference\_t, 737
- imagelistSize
  - unpack\_fms\_GetStoredImages\_t, 737
- imei\_no
  - unpack\_dms\_GetSerialNumbers\_t, 690
- imeiSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 677
- imeiSvnSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 677
- ImeiSvnString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 677
- imeisv\_svn
  - unpack\_dms\_GetSerialNumbers\_t, 690
- imgType
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 723
- ims.h, 1263
  - MAX\_NAME\_LEN, 1265
  - pack\_ims\_SLQSGetIMSSMSConfig, 1265
  - pack\_ims\_SLQSGetIMSUserConfig, 1265
  - pack\_ims\_SLQSGetIMSVoIPConfig, 1266
  - pack\_ims\_SLQSGetRegMgrConfig, 1266
  - pack\_ims\_SLQSGetSIPConfig, 1267
  - pack\_ims\_SLQSImsConfigIndicationRegister, 1267
  - pack\_ims\_SLQSSetIMSSMSConfig, 1268
  - pack\_ims\_SLQSSetIMSUserConfig, 1268
  - pack\_ims\_SLQSSetIMSVoIPConfig, 1269
  - pack\_ims\_SLQSSetRegMgrConfig, 1269
  - pack\_ims\_SLQSSetSIPConfig, 1269
  - unpack\_ims\_SLQSGetIMSSMSConfig, 1270
  - unpack\_ims\_SLQSGetIMSUserConfig, 1270
  - unpack\_ims\_SLQSGetIMSVoIPConfig, 1271
  - unpack\_ims\_SLQSGetRegMgrConfig, 1271
  - unpack\_ims\_SLQSGetSIPConfig, 1271

- unpack\_ims\_SLQSImsConfigIndicationRegister, 1272
- unpack\_ims\_SLQSImsConfigIndicationRegister\_t, 1265
- unpack\_ims\_SLQSRegMgrCfgCallBack\_ind, 1272
- unpack\_ims\_SLQSSIPCfgCallBack\_ind, 1274
- unpack\_ims\_SLQSSMSCfgCallBack\_ind, 1274
- unpack\_ims\_SLQSSetIMSSMSConfig, 1272
- unpack\_ims\_SLQSSetIMSUserConfig, 1273
- unpack\_ims\_SLQSSetIMSVolPConfig, 1273
- unpack\_ims\_SLQSSetRegMgrConfig, 1273
- unpack\_ims\_SLQSSetSIPConfig, 1274
- unpack\_ims\_SLQSUserCfgCallBack\_ind, 1275
- unpack\_ims\_SLQSVolPCfgCallBack\_ind, 1275
- ims\_AMRModelInfo, 84
  - amrMode, 84
  - TlvPresent, 85
- ims\_AMROctAlgnInfo, 85
  - amrOctAlgn, 85
  - TlvPresent, 85
- ims\_AMRWBModelInfo, 85
  - amrWBMode, 86
  - TlvPresent, 86
- ims\_AMRWBOctAlgnInfo, 86
  - amrWBOctAlgn, 86
  - TlvPresent, 86
- ims\_CSCFPortNameInfo, 86
  - cscfPortName, 87
  - TlvPresent, 87
- ims\_EnabAMRWBInfo, 87
  - amrWBEnable, 87
  - TlvPresent, 87
- ims\_EnabSCRAMRInfo, 87
  - scrAmrEnable, 88
  - TlvPresent, 88
- ims\_EnabSCRAMRWBInfo, 88
  - scrAmrWBEnable, 88
  - TlvPresent, 88
- ims\_IMSDomainInfo, 88
  - imsDomainName, 89
  - TlvPresent, 89
- ims\_IMSTestModelInfo, 89
  - imsTestMode, 89
  - TlvPresent, 89
- ims\_MinSessExpInfo, 89
  - minSessExp, 90
  - TlvPresent, 90
- ims\_PCSCFPortInfo, 90
  - priCSCFPort, 90
  - TlvPresent, 90
- ims\_PhCtxtURIInfo, 90
  - PhCtxtURI, 90
  - TlvPresent, 91
- ims\_RTPRTCPInactTmrDurInfo, 92
  - InactTmr, 92
  - TlvPresent, 92
- ims\_RngBkTmrInfo, 91
  - RingBkTmr, 91
  - TlvPresent, 91
- ims\_RngTmrInfo, 91
  - RingTmr, 91
  - TlvPresent, 92
- ims\_SIPPortInfo, 93
  - SIPLocalPort, 93
  - TlvPresent, 93
- ims\_SIPRegnTmrInfo, 94
  - TlvPresent, 94
  - tmrSIPRegn, 94
- ims\_SMSFmtInfo, 94
  - smsFormat, 94
  - TlvPresent, 94
- ims\_SMSolPNwInfo, 95
  - smsolPNW, 95
  - TlvPresent, 95
- ims\_SessDurInfo, 92
  - sessExp, 92
  - TlvPresent, 93
- ims\_SigCompEnInfo, 93
  - SigCompEn, 93
  - TlvPresent, 93
- ims\_SubscrTmrInfo, 95
  - subscrTmr, 95
  - TlvPresent, 95
- ims\_TmrT1Info, 95
  - TlvPresent, 96
  - tmrT1, 96
- ims\_TmrT2Info, 96
  - TlvPresent, 96
  - tmrT2, 96
- ims\_TmrTfInfo, 96
  - TlvPresent, 97
  - tmrTf, 97
- imsDomainName
  - ims\_IMSDomainInfo, 89
- ImFailErrCode
  - imsa\_IMSFailErrCodeTlv, 97
- ImRegErrCode
  - unpack\_imsa\_SLQSGetIMSARegStatus\_t, 754
- imsRegState
  - nas\_CommInfo, 190
- ImRegStatus
  - imsa\_NewIMSRegStatusInfo, 99
  - unpack\_imsa\_SLQSGetIMSARegStatus\_t, 754
- ImRegistered
  - imsa\_IMSRegStatusInfo, 98
- imsTestMode
  - ims\_IMSTestModelInfo, 89
- ImVoiceSupportLte
  - nas\_ImVoiceSupportLteTlv, 221
- imsa.h, 1275
  - MAX\_ERROR\_CODE\_LEN, 1277
  - pack\_imsa\_SLQSGetIMSARegStatus, 1277
  - pack\_imsa\_SLQSGetIMSAServiceStatus, 1277
  - pack\_imsa\_SLQSRegisterIMSARegStatus, 1278
  - unpack\_imsa\_SLQSGetIMSARegStatus, 1278
  - unpack\_imsa\_SLQSGetIMSAServiceStatus, 1279

- unpack\_imsa\_SLQSImsaPdpStatusCallBack\_ind, [1279](#)
- unpack\_imsa\_SLQSImsaRatStatusCallBack\_ind, [1279](#)
- unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind, [1280](#)
- unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind, [1280](#)
- unpack\_imsa\_SLQSRegisterIMSAIndication, [1281](#)
- unpack\_imsa\_SLQSRegisterIMSAIndication\_t, [1277](#)
- imsa\_IMSFailErrCodeTlv, [97](#)
  - ImsFailErrCode, [97](#)
  - TlvPresent, [97](#)
- imsa\_IMSRegStatusErrorCodeInfo, [97](#)
  - ErrorCode, [98](#)
  - TlvPresent, [98](#)
- imsa\_IMSRegStatusInfo, [98](#)
  - ImsRegistered, [98](#)
  - TlvPresent, [98](#)
- imsa\_NewIMSRegStatusInfo, [99](#)
  - ImsRegStatus, [99](#)
  - TlvPresent, [99](#)
- imsa\_RatHandoverStatusInfo, [99](#)
  - ErrorCodeData, [100](#)
  - ErrorCodeLen, [100](#)
  - RatHandoverStatus, [100](#)
  - SourceRAT, [100](#)
  - TargetRAT, [100](#)
  - TlvPresent, [100](#)
- imsa\_SmsRatInfo, [100](#)
  - SmsRatVal, [100](#)
  - TlvPresent, [101](#)
- imsa\_SmsSvcStatusInfo, [101](#)
  - SmsSvcStatus, [101](#)
  - TlvPresent, [101](#)
- imsa\_UtRatInfo, [101](#)
  - TlvPresent, [102](#)
  - UtRatVal, [102](#)
- imsa\_UtSvcStatusInfo, [102](#)
  - TlvPresent, [102](#)
  - UtSvcStatus, [102](#)
- imsa\_VoipRatInfo, [102](#)
  - TlvPresent, [103](#)
  - VoipRatVal, [103](#)
- imsa\_VoipSvcStatusInfo, [103](#)
  - TlvPresent, [103](#)
  - VoipSvcStatus, [103](#)
- imsa\_VtRatInfo, [103](#)
  - TlvPresent, [104](#)
  - VtRatVal, [104](#)
- imsa\_VtSvcStatusInfo, [104](#)
  - TlvPresent, [104](#)
  - VtSvcStatus, [104](#)
- imsi
  - unpack\_dms\_GetIMSI\_t, [682](#)
- imsi\_11\_12
  - nas\_CDMASysInfoExt, [185](#)
- imsiM1112
  - nas\_minBasedIMSI, [255](#)
- imsiMS1
  - nas\_minBasedIMSI, [255](#)
- imsiMS2
  - nas\_minBasedIMSI, [255](#)
- imsiT1112
  - nas\_trueIMSI, [335](#)
- imsiTS1
  - nas\_trueIMSI, [335](#)
- imsiTS2
  - nas\_trueIMSI, [335](#)
- imsiTaddrNum
  - nas\_trueIMSI, [335](#)
- InUse
  - nas\_QmiNas3GppNetworkInfo, [292](#)
- InactTmr
  - ims\_RTPRTCPInactTmrDurInfo, [92](#)
- includes\_pcs\_digit
  - nas\_QmisNasPcsDigit, [294](#)
- index
  - pack\_wds\_GetMobileIPProfile\_t, [558](#)
  - pack\_wds\_SetDefaultProfileNum\_t, [563](#)
  - pack\_wds\_SetMobileIPProfile\_t, [566](#)
  - uim\_UIMGetIndex, [651](#)
  - unpack\_qos\_swiQosFilter\_t, [921](#)
  - unpack\_qos\_swiQosFlow\_t, [925](#)
  - unpack\_wds\_GetDefaultProfileNum\_t, [1055](#)
- index1xPri
  - uim\_cardStatus, [629](#)
- index1xSec
  - uim\_cardStatus, [629](#)
- indexGwPri
  - uim\_cardStatus, [629](#)
- indexGwSec
  - uim\_cardStatus, [629](#)
- Info
  - unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t, [875](#)
  - unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t, [878](#)
- InfoInterfreq
  - nas\_LTEInfoInterfreq, [232](#)
- injectEnable
  - loc\_accelAcceptReady, [148](#)
  - loc\_accelTempAcceptReady, [149](#)
  - loc\_gyroAcceptReady, [157](#)
  - loc\_gyroTempAcceptReady, [157](#)
- injectSensorDataStatus
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, [778](#)
- injectTimeSyncStatus
  - unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t, [779](#)
- insNmrCellInfo
  - nas\_GERANInfo, [206](#)
- instancesSize
  - unpack\_nas\_GetRFInfo\_t, [802](#)
- intermediate

- loc\_IntermediateRptStateTlv, 159
- interval
  - loc\_MinIntervalTlv, 163
  - pack\_pds\_SetPDSDDefaults\_t, 476
  - pack\_pds\_SetXTRAAutomaticDownload\_t, 478
  - pack\_wds\_SLQSSetWdsEventCallback\_t, 575
- intf
  - dms\_UimStatusTlv, 73
- lo
  - unpack\_nas\_SLQSGetSignalStrength\_t, 833
- io
  - hdrSSInfo, 83
  - nas\_SLQSSignalStrengthsInformation, 324
- ioDelta
  - nas\_SLQSSignalStrengthsIndReq, 322
- lono\_valid
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- ipAddress
  - pack\_wds\_SetDefaultProfile\_t, 562
- ipFamily
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 1087
- ipVersion
  - LibPackTFTIDParams, 144
- ipaddr
  - unpack\_wds\_GetDefaultProfile\_t, 1054
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- ipaddrv6
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- ipv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- ipv4GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- ipv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- ipv6AddressInfo, 104
  - IPAddressV6, 104
  - IPv6PrefixLen, 105
- ipv6GWAddress
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- is3GppNw
  - nas\_homeNwMNC3GppTlv, 220
- is856SysId
  - nas\_HDRSysInfo, 219
- is856SysIdValid
  - nas\_HDRSysInfo, 219
- is\_DataRate\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_EspSpi\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv4DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv4SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv4Tos\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv6DstAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv6Label\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv6SrcAddr\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_IPv6TrafCls\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_Id\_Available
  - unpack\_qos\_swiQosFilter\_t, 921
- is\_Jitter\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_Latency\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_LteBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, 711
  - unpack\_dms\_SLQSGetBandCapabilityExt\_t, 715
- is\_LteQci\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_MaxAllowedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_MinPolicedPktSz\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_NxtHdrProto\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_PktErrRate\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_Precedence\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_ProfileId3GPP2\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_RxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, 910
- is\_TCPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_TCPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_TdsBandCapability\_Available
  - unpack\_dms\_SLQSGetBandCapability\_t, 711
  - unpack\_dms\_SLQSGetBandCapabilityExt\_t, 715
- is-TokenBucket\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_TrafficClass\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_TranDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_TranSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_TxQFlowGranted\_Available
  - unpack\_qos\_QosFlowInfo\_t, 910
- is\_UDPDstPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_UDPSrcPort\_Available
  - unpack\_qos\_swiQosFilter\_t, 922
- is\_val\_3GPP2Pri\_Available

- unpack\_qos\_swiQosFlow\_t, 925
- is\_val\_3GPPImCn\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_val\_3GPPResResidualBER\_Available
  - unpack\_qos\_swiQosFlow\_t, 925
- is\_val\_3GPPSigInd\_Available
  - unpack\_qos\_swiQosFlow\_t, 926
- is\_val\_3GPPTraHdlPri\_Available
  - unpack\_qos\_swiQosFlow\_t, 926
- isEmpty
  - voice\_getAllCallInformation, 1142
- isInTraffic
  - nas\_txInfo, 335
- isModByCC
  - voice\_SUPSInfo, 1155
- isNewFlow
  - unpack\_qos\_QosFlowInfoState\_t, 911
- isPrefDataPath
  - nas\_GSMSrvStatusInfo, 210
  - nas\_NR5GSerStatTlv, 267
  - nas\_NR5GSrvStatusTlv, 268
  - nas\_SrvStatusInfo, 327
- isRadioTuned
  - nas\_rxInfo, 314
  - nas\_RxSigInfo, 315
- isSysForbidden
  - nas\_detailSvcInfo, 196
  - nas\_NR5GSysInfoTlv, 271
  - nas\_sysInfoCommon, 329
- isSysForbiddenValid
  - nas\_NR5GSysInfoTlv, 271
  - nas\_sysInfoCommon, 329
- isSysPriMatch
  - nas\_CDMASysInfo, 184
  - nas\_HDRSysInfo, 219
- isSysPriMatchValid
  - nas\_CDMASysInfo, 184
  - nas\_HDRSysInfo, 219
- IsVoiceEnabled
  - pack\_dms\_SetCustFeature\_t, 385
  - unpack\_dms\_GetCustFeature\_t, 667
- iseUICC
  - uim\_physlotInfo, 639
- Item
  - pack\_audio\_SLQSGetAudioPathConfig\_t, 373
  - pack\_audio\_SLQSGetAudioVolTLBConfig\_t, 374
  - pack\_audio\_SLQSSetAudioVolTLBConfig\_t, 378
- Jitter
  - unpack\_qos\_swiQosFlow\_t, 926
- jtagAccessAllowed
  - unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, 966
- KeyExchange
  - nas\_protocolSubtypeElement, 290
- LITE\_SYS\_SRV\_DOMAIN\_CAMPED
  - nas.h, 1333
- LITE\_SYS\_SRV\_DOMAIN\_CS\_ONLY
  - nas.h, 1333
- LITE\_SYS\_SRV\_DOMAIN\_CS\_PS
  - nas.h, 1333
- LITE\_SYS\_SRV\_DOMAIN\_NO\_SRV
  - nas.h, 1333
- LITE\_SYS\_SRV\_DOMAIN\_PS\_ONLY
  - nas.h, 1333
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATA
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATA\_CIRCUITS-  
YNC
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATA\_CIRCUITS-  
NC
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_FAX
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_NONE
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_PADACCESS
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_SMS
  - voice.h, 1580
- LITE\_VOICE\_SUPS\_SRV\_CLASS\_VOICE
  - voice.h, 1580
- LITEQMI\_QMI\_CBK\_PARAM\_NOCHANGE
  - sms.h, 1474
- LITEQMI\_QMI\_CBK\_PARAM\_RESET
  - sms.h, 1474
- LITEQMI\_QMI\_CBK\_PARAM\_SET
  - sms.h, 1474
- LBPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- LITE\_MAX\_PCOID\_LIST
  - wds.h, 1614
- LMCERTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, 1072
- LOCEVENTMASKGNSSSVINFO
  - loc.h, 1299
- LOCEVENTMASKNMEA
  - loc.h, 1300
- LOCEVENTMASKWIFIREQ
  - loc.h, 1301
- LPCSTR
  - SwiDataTypes.h, 1517
- LTEAttachProfile
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1073
- LTEAttachProfileList
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1074
- LTEAttachProfileListLen
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 573
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1074
- LTETBandPref
  - nas\_LTETBandPrefTlv, 225



- NASLTEBandPreferenceTlv, 356
- LTECphyCAInfo
  - unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 816
- LTERSRPThreshListLen
  - nas\_LTERSRPThresh, 245
- LTERSRQThreshListLen
  - nas\_LTERSRQThresh, 245
- LTERSSIThreshListLen
  - nas\_LTERSSIThresh, 246
- LTESNRThresListLen
  - nas\_LTESNRThresh, 249
- LTESNRThreshListLen
  - nas\_LTESNRThreshold, 249
- LTESSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 865
- LTEWCDMACellInfo
  - nas\_LTEInfoNeighboringWCDMA, 236
- Lac
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- lac
  - nas\_GERANInfo, 206
  - nas\_GSMsSysInfo, 213
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 271
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
  - nas\_WCDMASysInfo, 351
- lac1
  - nas\_OperatorPLMNData, 278
- lac2
  - nas\_OperatorPLMNData, 278
- lacValid
  - nas\_GSMsSysInfo, 213
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 271
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_WCDMASysInfo, 351
- language
  - sms\_CDMABroadcastConfig, 596
- lastCallDataBearerTech
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1079
- lastCallDataBearerTechnology
  - unpack\_wds\_SLQSGetDataBearerTechnology\_t, 1077
- lastCallRXOKBytesCnt
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1079
- lastCallTXOKBytesCnt
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1079
- LastErrCode
  - nas\_DataStatusDetail, 194
- Latency
  - unpack\_qos\_swiQosFlow\_t, 926
- Latitude
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- latitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, 418
- leapSeconds
  - nas\_qaQmi3Gpp2TimeZone, 291
- leaseState
  - wds\_DHCPLeaseStateTlv, 1163
- len
  - loc\_BdsSVInfo, 151
  - loc\_SVInfo, 168
  - unpack\_nas\_GetSignalStrengths\_t, 806
- length
  - sMSCAddressInfo, 608
  - sMSEtwsMessageInfo, 609
  - sMSTransferRouteMTMessageInfo, 613
  - uim\_readRecordInfo, 640
  - uim\_readTransparentInfo, 642
- Level
  - pack\_swiaudio\_SLQSSetM2MAudioVolume\_t, 502
  - unpack\_swiaudio\_SLQSGetM2MAudioVolume\_t, 952
- LibPackGPRSRequestedQoS, 105
  - delayClass, 105
  - meanThroughputClass, 105
  - peakThroughputClass, 105
  - precedenceClass, 105
  - reliabilityClass, 105
- LibPackPCOIDList, 105
  - PcoList, 106
- LibPackPDNThrottleTimer, 106
  - ThrottleTimer, 106
- LibPackProfileMnc, 140
  - MNC, 141
  - PCSFlag, 141
- LibPackQosClassID, 141
  - gDIBitRate, 142
  - gUIBitRate, 142
  - maxDIBitRate, 142
  - maxUIBitRate, 142
  - QCI, 142
- LibPackTFTIDParams, 142
  - destPortRangeEnd, 143
  - destPortRangeStart, 143
  - eValid, 143
  - filterId, 143
  - flowLabel, 143
  - IPSECSPi, 143
  - ipVersion, 144
  - nextHeader, 144
  - pSourceIP, 144
  - sourceIPMask, 144
  - srcPortRangeEnd, 144
  - srcPortRangeStart, 144
  - tosMask, 144
- LibPackUMTSQoS, 144
  - deliveryErrSDU, 146
  - grntDownlinkBitrate, 146
  - grntUplinkBitrate, 146
  - maxDownlinkBitrate, 146
  - maxSDUSize, 146
  - maxUplinkBitrate, 146

- qosDeliveryOrder, 146
- resBerRatio, 146
- sduErrorRatio, 146
- trafficClass, 146
- trafficPriority, 146
- transferDelay, 146
- LibPackUMTSReqQoSsigInd, 147
  - SigInd, 147
  - UMTSReqQoS, 147
- LibPackprofile\_3GPP, 129
  - pAPNClass, 134
  - pAPNDisabledFlag, 134
  - pAPNName, 134
  - pAPNnameSize, 134
  - pAddrAllocPref, 134
  - pAuthenticationPref, 134
  - pGPRSMinimumQoS, 134
  - pGPRSRequestedQos, 134
  - pIPv4AddrPref, 134
  - pIPv6AddPref, 134
  - plmCnFlag, 134
  - pPDNInactivTimeout, 134
  - pPDPTYPE, 135
  - pPassword, 134
  - pPasswordSize, 134
  - pPcscfAddrUsingDhcp, 134
  - pPcscfAddrUsingPCO, 134
  - pPdpAccessConFlag, 134
  - pPdpContext, 134
  - pPdpDataCompType, 134
  - pPdpHdrCompType, 134
  - pPriDNSIPv4AddPref, 135
  - pPriDNSIPv6addpref, 135
  - pPrimaryID, 135
  - pProfilename, 135
  - pProfilenameSize, 135
  - pQosClassID, 135
  - pSecDNSIPv4AddPref, 135
  - pSecDNSIPv6addpref, 135
  - pSecondaryFlag, 135
  - pSupportEmergencyCalls, 135
  - pTFTID1Params, 135
  - pTFTID2Params, 135
  - pUMTSMinQoS, 135
  - pUMTSMinQosSigInd, 135
  - pUMTSReqQoS, 135
  - pUMTSReqQoSsigInd, 135
  - pUsername, 135
  - pUsernameSize, 135
- LibPackprofile\_3GPP2, 135
  - pAPNClass3GPP2, 139
  - pAPNEnabled3GPP2, 139
  - pAllowLinger, 139
  - pApnString, 139
  - pApnStringSize, 139
  - pAppPriority, 140
  - pAppType, 140
  - pAuthPassword, 140
  - pAuthPassword\_tSize, 140
  - pAuthProtocol, 140
  - pAuthRetryCount, 140
  - pAuthTimeout, 140
  - pDataMode, 140
  - pDataRate, 140
  - plpcpAckTimeout, 140
  - plpcpCreqRetryCount, 140
  - plsPcscfAddressNedded, 140
  - pLcpAckTimeout, 140
  - pLcpCreqRetryCount, 140
  - pNegoDnsSrvrPref, 140
  - pPDNInactivTimeout3GPP2, 140
  - pPdnType, 140
  - pPppSessCloseTimer1x, 140
  - pPppSessCloseTimerDO, 140
  - pPriV6DnsAddress, 140
  - pPrimaryV4DnsAddress, 140
  - pRATType, 140
  - pSecV6DnsAddress, 140
  - pSecondaryV4DnsAddress, 140
  - pUserId, 140
  - pUserIdSize, 140
- libSDP\_BuildImagesPreferenceRequest
  - lite-fw.h, 1284
- libSDP\_CalculateImageMask
  - lite-fw.h, 1284
- libSDP\_CheckValidFirmwareInfo
  - lite-fw.h, 1284
- libSDP\_DownloadFW
  - lite-fw.h, 1284
- libSDP\_ExtractFirmwareParametersByPath
  - lite-fw.h, 1284
- libSDP\_FirmwareInfo
  - lite-fw.h, 1285
- libSDP\_GetModelFamily
  - lite-fw.h, 1284
- libSDP\_GetVersion
  - lite-fw.h, 1284
- libSDP\_getFileType
  - lite-fw.h, 1284
- LibpackProfile3GPP, 106
  - pAPNClass, 112
  - pAPNDisabledFlag, 112
  - pAPNName, 112
  - pAPNnameSize, 112
  - pAddrAllocPref, 112
  - pAuthenticationPref, 112
  - pGPRSMinimumQoS, 112
  - pGPRSRequestedQos, 112
  - pIPv4AddrPref, 112
  - pIPv6AddPref, 112
  - plmCnFlag, 112
  - pPDNInactivTimeout, 112
  - pPDPTYPE, 112
  - pPassword, 112
  - pPasswordSize, 112
  - pPcscfAddrUsingDhcp, 112



- pPcscfAddrUsingPCO, 112
- pPdpAccessConFlag, 112
- pPdpContext, 112
- pPdpDataCompType, 112
- pPdpHdrCompType, 112
- pPriDNSIPv4AddPref, 112
- pPriDNSIPv6addpref, 112
- pPrimaryID, 112
- pProfileName, 112
- pProfileNameSize, 112
- pQoSClassID, 112
- pSecDNSIPv4AddPref, 112
- pSecDNSIPv6addpref, 113
- pSecondaryFlag, 113
- pSupportEmergencyCalls, 113
- pTFTID1Params, 113
- pTFTID2Params, 113
- pUMTSMInQoS, 113
- pUMTSMInQoSSigInd, 113
- pUMTSReqQoS, 113
- pUMTSReqQoSSigInd, 113
- pUsername, 113
- pUsernameSize, 113
- LibpackProfile3GPP2, 113
  - pAPNClass3GPP2, 118
  - pAPNEnabled3GPP2, 118
  - pAllowLinger, 118
  - pApnString, 118
  - pApnStringSize, 118
  - pAppPriority, 118
  - pAppType, 118
  - pAuthPassword, 118
  - pAuthPasswordSize, 118
  - pAuthProtocol, 118
  - pAuthRetryCount, 118
  - pAuthTimeout, 118
  - pDataMode, 118
  - pDataRate, 118
  - pIpcpAckTimeout, 118
  - pIpcpCreqRetryCount, 118
  - pIscPcscfAddressNedded, 118
  - pLcpAckTimeout, 118
  - pLcpCreqRetryCount, 118
  - pNegoDnsSrvrPref, 118
  - pPDNInactivTimeout3GPP2, 118
  - pPdnType, 118
  - pPppSessCloseTimer1x, 118
  - pPppSessCloseTimerDO, 119
  - pPriV6DnsAddress, 119
  - pPrimaryV4DnsAddress, 119
  - pRATType, 119
  - pSecV6DnsAddress, 119
  - pSecondaryV4DnsAddress, 119
  - pUserId, 119
  - pUserIdSize, 119
- LibpackProfile3GPPV2, 119
  - pAPNBearer, 127
  - pAPNClass, 127
  - pAPNDisabledFlag, 127
  - pAPNName, 127
  - pAPNnameSize, 127
  - pAddrAllocPref, 127
  - pAppUserData, 127
  - pAuthenticationPref, 127
  - pClatFlag, 127
  - pDnsWithDHCPFlag, 127
  - pGPRSMInQoS, 127
  - pGPRSRequestedQoS, 127
  - pIPv6DelegFlag, 128
  - pIPv4AddrPref, 128
  - pIPv6AddPref, 128
  - pIWLANtoLTEHandoverFlag, 128
  - pImCnFlag, 128
  - pLTetoIWLANHandoverFlag, 128
  - pLteRoamPDPTType, 128
  - pMaxPDN, 128
  - pMaxPDNTimer, 128
  - pMcc, 128
  - pMnc, 128
  - pMsisdnFlag, 128
  - pOperatorPCOID, 128
  - pOverridePDPTType, 128
  - pPCOIDList, 128
  - pPDNDisconnectWaitTimer, 128
  - pPDNInactivTimeout, 128
  - pPDNThrottleTimer, 128
  - pPDNWaitTimer, 128
  - pPDPTtype, 128
  - pPassword, 128
  - pPasswordSize, 128
  - pPcscfAddrUsingDhcp, 128
  - pPcscfAddrUsingPCO, 128
  - pPdpAccessConFlag, 128
  - pPdpContext, 128
  - pPdpDataCompType, 128
  - pPdpHdrCompType, 128
  - pPersistFlag, 129
  - pPriDNSIPv4AddPref, 129
  - pPriDNSIPv6addpref, 129
  - pPrimaryID, 129
  - pProfileName, 129
  - pProfileNameSize, 129
  - pQoSClassID, 129
  - pRoamDisallowFlag, 129
  - pSecDNSIPv4AddPref, 129
  - pSecDNSIPv6addpref, 129
  - pSecondaryFlag, 129
  - pSupportEmergencyCalls, 129
  - pTFTID1Params, 129
  - pTFTID2Params, 129
  - pUMTSMInQoS, 129
  - pUMTSMInQoSSigInd, 129
  - pUMTSReqQoS, 129
  - pUMTSReqQoSSigInd, 129
  - pUmtsRoamPDPTType, 129
  - pUsername, 129

- pUsernameSize, [129](#)
- libsdp\_SetReadBlockSize
  - lite-fw.h, [1284](#)
- libsdp\_set\_log\_func
  - lite-fw.h, [1284](#)
- libsdplogger
  - lite-fw.h, [1285](#)
- lineValue
  - pack\_voice\_SLQSVoiceALSSelectLine\_t, [538](#)
- linkage
  - altSrcInfo\_t, [39](#)
- list\_type
  - DMScustSettingList, [75](#)
  - DMSgetCustomInput, [77](#)
  - pack\_dms\_GetCustFeaturesV2\_t, [381](#)
- listEntries
  - FMSPrefImageList, [82](#)
- listSize
  - FMSImageList, [81](#)
  - FMSPrefImageList, [82](#)
- lite-fw.h
  - eFILE\_TYPE\_CAR\_PRI, [1287](#)
  - eFILE\_TYPE\_COMPO\_PRI, [1287](#)
  - eFILE\_TYPE\_NONE, [1287](#)
  - eFILE\_TYPE\_OEM\_PRI, [1287](#)
  - eFIREHOSE\_ERR\_SECBOOT\_INVALID\_CERT\_CHAIN, [1287](#)
  - eFW\_TYPE\_CWE, [1287](#)
  - eFW\_TYPE\_CWE\_NVU, [1287](#)
  - eFW\_TYPE\_INVALID, [1287](#)
  - eFW\_TYPE\_MBN, [1287](#)
  - eFW\_TYPE\_MBN\_GOBI, [1287](#)
  - eFW\_TYPE\_NVU, [1287](#)
  - eFW\_TYPE\_SPK, [1287](#)
  - eIMAGE\_TYPE\_ANY, [1288](#)
  - eIMAGE\_TYPE\_APPL, [1288](#)
  - eIMAGE\_TYPE\_APPS, [1288](#)
  - eIMAGE\_TYPE\_BOOT, [1288](#)
  - eIMAGE\_TYPE\_FILE, [1288](#)
  - eIMAGE\_TYPE\_INVALID, [1288](#)
  - eIMAGE\_TYPE\_KEYS, [1288](#)
  - eIMAGE\_TYPE\_MAX, [1288](#)
  - eIMAGE\_TYPE\_MIN, [1288](#)
  - eIMAGE\_TYPE\_MODM, [1288](#)
  - eIMAGE\_TYPE\_SPKG, [1288](#)
  - eIMAGE\_TYPE\_USER, [1288](#)
  - eModel\_9X15, [1288](#)
  - eModel\_9X30, [1288](#)
  - eModel\_9x06, [1288](#)
  - eModel\_9x07, [1288](#)
  - eModel\_Unknown, [1288](#)
  - eModel\_WP9X15, [1288](#)
  - eQDL\_HW\_FLOW\_DISABLE, [1288](#)
  - eQDL\_HW\_FLOW\_ENABLE, [1288](#)
  - eQDL\_HW\_FLOW\_INIT, [1288](#)
  - eQDL\_HW\_FLOW\_Unknown, [1288](#)
  - eQDL\_MODE\_INIT, [1288](#)
  - eQDL\_MODE\_TTYUSB, [1288](#)
  - eQDL\_MODE\_UART0, [1288](#)
  - eQDL\_MODE\_Unknown, [1288](#)
  - eSDP\_FWDWL\_ERR\_END, [1287](#)
  - eSDP\_FWDWL\_ERR\_FAIL, [1287](#)
  - eSDP\_FWDWL\_ERR\_FW\_UPGRADE, [1287](#)
  - eSDP\_FWDWL\_ERR\_FW\_VERSION\_FAIL, [1287](#)
  - eSDP\_FWDWL\_ERR\_GENERAL, [1287](#)
  - eSDP\_FWDWL\_ERR\_INVALID\_DEV, [1287](#)
  - eSDP\_FWDWL\_ERR\_INVALID\_PATH, [1287](#)
  - eSDP\_FWDWL\_ERR\_PATH\_NOT\_SPECIFIED, [1287](#)
  - eSDP\_FWDWL\_ERR\_PATH\_TOO\_LONG, [1287](#)
  - eSDP\_FWDWL\_ERR\_PRI\_FAIL, [1287](#)
  - eSDP\_FWDWL\_ERR\_SDK, [1287](#)
  - eSDP\_FWDWL\_ERR\_SDP\_TIMEOUT, [1287](#)
  - eSDP\_FWDWL\_ERR\_SET\_CBK, [1287](#)
  - eSDP\_FWDWL\_ERR\_TIMEOUT, [1287](#)
  - eSDP\_FWDWL\_SUCCESS, [1287](#)
- lite-fw.h, [1281](#)
  - FIRMWARE\_BCHDATESIZE, [1284](#)
  - IMG\_MASK\_CLEAR, [1284](#)
  - IMG\_MASK\_GENERIC, [1284](#)
  - IMG\_MASK\_MDM, [1284](#)
  - IMG\_MASK\_PRI, [1284](#)
  - libSDP\_BuildImagesPreferenceRequest, [1284](#)
  - libSDP\_CalculateImageMask, [1284](#)
  - libSDP\_CheckValidFirmwareInfo, [1284](#)
  - libSDP\_DownloadFW, [1284](#)
  - libSDP\_ExtractFirmwareParametersByPath, [1284](#)
  - libSDP\_FirmwareInfo, [1285](#)
  - libSDP\_GetModelFamily, [1284](#)
  - libSDP\_GetVersion, [1284](#)
  - libSDP\_getFileType, [1284](#)
  - libsdp\_SetReadBlockSize, [1284](#)
  - libsdp\_set\_log\_func, [1284](#)
  - libsdplogger, [1285](#)
  - litefw\_BuildImagesPreferenceRequest, [1289](#)
  - litefw\_CalculateImageMask, [1289](#)
  - litefw\_CheckValidFirmwareInfo, [1289](#)
  - litefw\_DownloadFW, [1289](#)
  - litefw\_ExtractFirmwareParametersByPath, [1290](#)
  - litefw\_ExtractFirmwarePartNoByPath, [1290](#)
  - litefw\_FirmwareFileInfo, [1285](#)
  - litefw\_FirmwareInfo, [1285](#)
  - litefw\_FirmwarePartNo, [1286](#)
  - litefw\_Fw\_Type, [1287](#)
  - litefw\_GetModelFamily, [1291](#)
  - litefw\_GetQTLDownloadMode, [1292](#)
  - litefw\_GetQTLHWFlowControl, [1292](#)
  - litefw\_GetVersion, [1292](#)
  - litefw\_Models, [1288](#)
  - litefw\_QDL\_FLOW\_CONTROLS, [1288](#)
  - litefw\_QDL\_MODEs, [1288](#)
  - litefw\_SLQSGetFirmwareFileInfo, [1293](#)
  - litefw\_SetQTLDownloadMode, [1292](#)
  - litefw\_SetQTLHWFlowControl, [1293](#)
  - litefw\_SetReadBlockSize, [1293](#)
  - litefw\_fileimgtype, [1286](#)

- litefw\_fwdwl\_error\_codes, [1287](#)
- litefw\_getFileType, [1291](#)
- litefw\_imagetype, [1287](#)
- litefw\_logsenable, [1292](#)
- litefw\_set\_log\_func, [1292](#)
- litefw\_switch\_9x07\_to\_downloadmode, [1294](#)
- litefw\_switch\_to\_BootHoldMode, [1294](#)
- litefwlogger, [1286](#)
- liteQmiDataBearerMasks
  - wds.h, [1616](#)
- liteServiceClassInformation
  - voice.h, [1580](#)
- litefw\_BuildImagesPreferenceRequest
  - lite-fw.h, [1289](#)
- litefw\_CalculatelmageMask
  - lite-fw.h, [1289](#)
- litefw\_CheckValidFirmwareInfo
  - lite-fw.h, [1289](#)
- litefw\_DownloadFW
  - lite-fw.h, [1289](#)
- litefw\_ExtractFirmwareParametersByPath
  - lite-fw.h, [1290](#)
- litefw\_ExtractFirmwarePartNoByPath
  - lite-fw.h, [1290](#)
- litefw\_FirmwareFileInfo
  - lite-fw.h, [1285](#)
- litefw\_FirmwareInfo
  - lite-fw.h, [1285](#)
- litefw\_FirmwarePartNo
  - lite-fw.h, [1286](#)
- litefw\_Fw\_Type
  - lite-fw.h, [1287](#)
- litefw\_GetModelFamily
  - lite-fw.h, [1291](#)
- litefw\_GetQTLDownloadMode
  - lite-fw.h, [1292](#)
- litefw\_GetQTLHWFlowControl
  - lite-fw.h, [1292](#)
- litefw\_GetVersion
  - lite-fw.h, [1292](#)
- litefw\_Models
  - lite-fw.h, [1288](#)
- litefw\_QDL\_FLOW\_CONTROLS
  - lite-fw.h, [1288](#)
- litefw\_QDL\_MODES
  - lite-fw.h, [1288](#)
- litefw\_SLQSGetFirmwareFileInfo
  - lite-fw.h, [1293](#)
- litefw\_SetQTLDownloadMode
  - lite-fw.h, [1292](#)
- litefw\_SetQTLHWFlowControl
  - lite-fw.h, [1293](#)
- litefw\_SetReadBlockSize
  - lite-fw.h, [1293](#)
- litefw\_fileimgtype
  - lite-fw.h, [1286](#)
- litefw\_fwdwl\_error\_codes
  - lite-fw.h, [1287](#)
- litefw\_getFileType
  - lite-fw.h, [1291](#)
- litefw\_imagetype
  - lite-fw.h, [1287](#)
- litefw\_logsenable
  - lite-fw.h, [1292](#)
- litefw\_set\_log\_func
  - lite-fw.h, [1292](#)
- litefw\_switch\_9x07\_to\_downloadmode
  - lite-fw.h, [1294](#)
- litefw\_switch\_to\_BootHoldMode
  - lite-fw.h, [1294](#)
- litefwlogger
  - lite-fw.h, [1286](#)
- liteqmi\_GetVersion
  - common.h, [1198](#)
- liteqmi\_helper\_decode7bitAsciiEncString
  - common.h, [1198](#)
- liteqmi\_log
  - common.h, [1198](#)
- loc.h
  - eQMI\_LOC\_SESS\_STATUS\_FAILURE, [1301](#)
  - eQMI\_LOC\_SESS\_STATUS\_IN\_PROGRESS, [1301](#)
  - eQMI\_LOC\_SESS\_STATUS\_SUCCESS, [1301](#)
  - eQMI\_LOC\_SESS\_STATUS\_TIMEOUT, [1301](#)
- loc.h, [1294](#)
  - LOCEVENTMASKNMEA, [1300](#)
  - LOCEVENTMASKWIFIREQ, [1301](#)
  - MAX\_TEMP\_DATA\_LEN, [1301](#)
  - pack\_loc\_DeleteAssistData, [1301](#)
  - pack\_loc\_EventRegister, [1302](#)
  - pack\_loc\_GetFixCriteria, [1302](#)
  - pack\_loc\_SLQSLOCGetBestAvailPos, [1303](#)
  - pack\_loc\_SLQSLOCGetOpMode, [1304](#)
  - pack\_loc\_SLQSLOCGetServer, [1304](#)
  - pack\_loc\_SLQSLOCInjectPosition, [1304](#)
  - pack\_loc\_SLQSLOCInjectSensorData, [1305](#)
  - pack\_loc\_SLQSLOCInjectUTCTime, [1305](#)
  - pack\_loc\_SLQSLOCSetCradleMountConfig, [1306](#)
  - pack\_loc\_SLQSLOCSetServer, [1306](#)
  - pack\_loc\_SetExtPowerState, [1302](#)
  - pack\_loc\_SetOperationMode, [1303](#)
  - pack\_loc\_Start, [1306](#)
  - pack\_loc\_Stop, [1307](#)
  - unpack\_loc\_BestAvailPos\_Ind, [1307](#)
  - unpack\_loc\_CradleMountCallback\_Ind, [1308](#)
  - unpack\_loc\_DeleteAssistData, [1308](#)
  - unpack\_loc\_DeleteAssistData\_Ind, [1308](#)
  - unpack\_loc\_DeleteAssistData\_t, [1301](#)
  - unpack\_loc\_EngineState\_Ind, [1309](#)
  - unpack\_loc\_EventNMEA\_Ind, [1309](#)
  - unpack\_loc\_EventRegister, [1309](#)
  - unpack\_loc\_EventTimeSyncCallback\_Ind, [1310](#)
  - unpack\_loc\_FixCriteria\_Ind, [1310](#)
  - unpack\_loc\_GetFixCriteria, [1310](#)
  - unpack\_loc\_GetFixCriteria\_t, [1301](#)
  - unpack\_loc\_GetOpMode\_Ind, [1311](#)

- unpack\_loc\_GetServer\_Ind, 1311
- unpack\_loc\_GnssSvInfo\_Ind, 1311
- unpack\_loc\_InjectPositionCallback\_Ind, 1312
- unpack\_loc\_InjectSensorDataCallback\_Ind, 1312
- unpack\_loc\_InjectTimeSyncDataCallback\_Ind, 1312
- unpack\_loc\_InjectUTCTimeCallback\_Ind, 1313
- unpack\_loc\_PositionRpt\_Ind, 1313
- unpack\_loc\_SLQSLOCGetBestAvailPos, 1315
- unpack\_loc\_SLQSLOCGetOpMode, 1316
- unpack\_loc\_SLQSLOCGetServer, 1316
- unpack\_loc\_SLQSLOCGetServer\_t, 1301
- unpack\_loc\_SLQSLOCInjectPosition, 1316
- unpack\_loc\_SLQSLOCInjectPosition\_t, 1301
- unpack\_loc\_SLQSLOCInjectSensorData, 1317
- unpack\_loc\_SLQSLOCInjectSensorData\_t, 1301
- unpack\_loc\_SLQSLOCInjectUTCTime, 1317
- unpack\_loc\_SLQSLOCInjectUTCTime\_t, 1301
- unpack\_loc\_SLQSLOCSetCradleMountConfig, 1317
- unpack\_loc\_SLQSLOCSetCradleMountConfig\_t, 1301
- unpack\_loc\_SLQSLOCSetServer, 1318
- unpack\_loc\_SensorStreamingCallback\_Ind, 1313
- unpack\_loc\_SetExtPowerConfig\_Ind, 1314
- unpack\_loc\_SetExtPowerState, 1314
- unpack\_loc\_SetOperationMode, 1314
- unpack\_loc\_SetOperationMode\_Ind, 1315
- unpack\_loc\_SetServer\_Ind, 1315
- unpack\_loc\_Start, 1318
- unpack\_loc\_Stop, 1318
- loc\_AppProviderInfoTlv, 149
  - name, 150
  - nameLen, 150
  - provider, 150
  - providerLen, 150
  - TlvPresent, 150
  - verValid, 150
  - version, 150
  - versionLen, 150
- loc\_BdsSV, 150
  - id, 150
  - mask, 150
- loc\_BdsSVInfo, 151
  - len, 151
  - pSV, 151
- loc\_CellDb, 151
  - mask, 152
- loc\_ClkInfo, 152
  - mask, 153
- loc\_FixCriteriaStatusTlv, 153
  - status, 154
  - TlvPresent, 154
- loc\_GnssData, 154
  - mask, 155
- loc\_HorAccuracyLvlTlv, 158
  - accuracy, 158
  - TlvPresent, 158
- loc\_IPv4Config, 159
  - IPv4Addr, 159
  - IPv4Port, 159
- loc\_IPv4Info, 159
  - address, 160
  - port, 160
  - TlvPresent, 160
- loc\_IPv6Config, 160
  - IPv6Addr, 160
  - IPv6Port, 160
- loc\_IPv6Info, 160
  - address, 161
  - port, 161
  - TlvPresent, 161
- loc\_IntermediateRptStateTlv, 158
  - intermediate, 159
  - TlvPresent, 159
- loc\_LocApplicationInfo, 161
  - appNameLength, 162
  - appProviderLength, 162
  - appVersionLength, 162
  - appVersionValid, 162
  - pAppName, 162
  - pAppProvider, 162
  - pAppVersion, 162
- loc\_MinIntervalTlv, 162
  - interval, 163
  - TlvPresent, 163
- loc\_SV, 166
  - id, 167
  - mask, 167
  - system, 167
- loc\_SVInfo, 167
  - len, 168
  - pSV, 168
- loc\_URLAddrInfo, 169
  - urlAddr, 169
- loc\_accelAcceptReady, 147
  - batchPerSec, 148
  - injectEnable, 148
  - samplesPerBatch, 148
- loc\_accelTempAcceptReady, 148
  - batchPerSec, 149
  - injectEnable, 149
  - samplesPerBatch, 149
- loc\_appProviderInfo
  - unpack\_loc\_FixCriteria\_Ind\_t, 772
- loc\_fixCriteriaStatus
  - unpack\_loc\_FixCriteria\_Ind\_t, 772
- loc\_gpsTime, 155
  - gpsTimeOfWeekMs, 156
  - gpsWeek, 156
- loc\_gyroAcceptReady, 156
  - batchPerSec, 157
  - injectEnable, 157
  - samplesPerBatch, 157
- loc\_gyroTempAcceptReady, 157
  - batchPerSec, 157

- injectEnable, [157](#)
- samplesPerBatch, [157](#)
- loc\_horAccuracy
  - unpack\_loc\_FixCriteria\_Ind\_t, [772](#)
- loc\_intermediateRptState
  - unpack\_loc\_FixCriteria\_Ind\_t, [772](#)
- loc\_minInterval
  - unpack\_loc\_FixCriteria\_Ind\_t, [772](#)
- loc\_precisionDilution, [163](#)
  - HDOP, [163](#)
  - PDOP, [163](#)
  - VDOP, [163](#)
- loc\_satelliteInfo, [163](#)
  - azimuth, [165](#)
  - elevation, [165](#)
  - gnssSvId, [165](#)
  - healthStatus, [165](#)
  - snr, [165](#)
  - svInfoMask, [165](#)
  - svListLen, [166](#)
  - svStatus, [166](#)
  - system, [166](#)
  - validMask, [166](#)
- loc\_sensorDataUsage, [166](#)
  - aidingIndicatorMask, [166](#)
  - usageMask, [166](#)
- loc\_svUsedforFix, [168](#)
  - gnssSvUsedList, [168](#)
  - gnssSvUsedList\_len, [168](#)
- loc\_urlAddr, [168](#)
  - address, [169](#)
  - TlvPresent, [169](#)
- localTimeOffset
  - nas\_qaQmi3Gpp2TimeZone, [291](#)
- logString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [723](#)
- logger
  - switype\_256bit.h, [1544](#)
- longName
  - nas\_PLMNNetworkNameData, [288](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [821](#)
- longNameCi
  - unpack\_nas\_SLQSGetPLMNName\_t, [821](#)
- longNameEn
  - unpack\_nas\_SLQSGetPLMNName\_t, [821](#)
- longNameLen
  - nas\_PLMNNetworkNameData, [288](#)
  - unpack\_nas\_SLQSGetPLMNName\_t, [821](#)
- longNameSB
  - unpack\_nas\_SLQSGetPLMNName\_t, [821](#)
- longNameSpareBits
  - nas\_PLMNNetworkNameData, [288](#)
- Longitude
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- longitude
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- loopbackMode
  - pack\_wds\_SLQSSSetLoopback\_t, [576](#)
- loopbackMultiplier
  - pack\_wds\_SLQSSSetLoopback\_t, [576](#)
- LpmFlag
  - unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, [703](#)
- LteBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, [711](#)
  - unpack\_dms\_SLQSGetBandCapabilityExt\_t, [715](#)
- LteBands
  - dms\_LteBandsSupport, [65](#)
- LteBandsSupport
  - unpack\_dms\_SLQSGetBandCapabilityExt\_t, [715](#)
- LteEarfcn
  - nas\_LteEarfcnInfo, [227](#)
- LteEmbmsCoverage
  - nas\_LteEmbmsCoverageTlv, [228](#)
- LteEmbmsTraceId
  - nas\_LteEmbmsTraceIdTlv, [228](#)
- LteEmmDI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- LteEmmUI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- LteEsmDI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- LteEsmUI
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- LteGsmCellInfo
  - nas\_LTEInfoNeighboringGSM, [235](#)
- LteInterEarfcnlen
  - nas\_LteEarfcnInfo, [227](#)
- LteM1BandPref
  - NASLteM1BandPrefTlv, [356](#)
- LteM1BandPref
  - nas\_LteM1BandPrefTlv, [237](#)
- LteNB1BandPref
  - NASLteNB1BandPrefTlv, [358](#)
- LteNb1BandPref
  - nas\_LteNb1BandPrefTlv, [239](#)
- LteOpMode
  - nas\_EdxCiotLteMode, [200](#)
  - nas\_LteOpMode, [241](#)
  - nas\_LteOpModeTlv, [242](#)
- LteOpModeLen
  - nas\_LteOpModeTlv, [242](#)
- LteOperationalMode
  - unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, [871](#)
- LteQci
  - unpack\_qos\_swiQosFlow\_t, [926](#)
- LteRegDomain
  - nas\_LteRegDomainTlv, [244](#)
- LteRsrpDelta
  - nas\_SLQSSignalStrengthsIndReq, [322](#)
- LteRsrpinfo
  - nas\_SLQSSignalStrengthsInformation, [324](#)
- LteSSInfo, [169](#)
  - rsrp, [170](#)
  - rsrq, [170](#)
  - rssi, [170](#)

- snr, [170](#)
- lteSnrDelta
  - nas\_SLQSSignalStrengthsIndReq, [322](#)
- lteSnrinfo
  - nas\_SLQSSignalStrengthsInformation, [324](#)
- LteVoiceDomain
  - nas\_LteVoiceDomainTlv, [252](#)
- ltersrp
  - unpack\_nas\_SLQSGetSignalStrength\_t, [833](#)
- ltesnr
  - unpack\_nas\_SLQSGetSignalStrength\_t, [833](#)
- lvi
  - unpack\_tmd\_SLQSTmdMitigationLviRptCallback\_ind\_t, [983](#)
- M1BandPTlv
  - NASQmiCbkNasSystemSelPrefInd, [368](#)
- m\_FwBuildId
  - CarrierImage\_t, [48](#)
- m\_FwImageId
  - CarrierImage\_t, [48](#)
- m\_PriBuildId
  - CarrierImage\_t, [48](#)
- m\_PrImageId
  - CarrierImage\_t, [49](#)
- m\_nCarrierId
  - CarrierImage\_t, [48](#)
- m\_nFolderId
  - CarrierImage\_t, [48](#)
- m\_nStorage
  - CarrierImage\_t, [48](#)
- MACIndex
  - nas\_NetworkStatEVDO, [263](#)
- MAX\_ATR\_LENGTH
  - uim.h, [1553](#)
- MAX\_BUILD\_ID\_LEN
  - dms.h, [1207](#)
- MAX\_ERROR\_CODE\_LEN
  - imsa.h, [1277](#)
- MAX\_ICCID\_LENGTH
  - uim.h, [1553](#)
- MAX\_MSE\_TWS\_MSG
  - sms.h, [1473](#)
- MAX\_NAME\_LEN
  - ims.h, [1265](#)
- MAX\_NO\_OF\_SLOTS
  - uim.h, [1553](#)
- MAX\_PHY\_SLOTS\_INFO
  - uim.h, [1553](#)
- MAX\_SENSOR\_DATA\_LEN
  - loc.h, [1301](#)
- MAX\_SLOTS\_STATUS
  - uim.h, [1553](#)
- MAX\_SMS\_LIST\_SIZE
  - sms.h, [1473](#)
- MAX\_TEMP\_DATA\_LEN
  - loc.h, [1301](#)
- MAXVOICEUSSDLENGTH
  - voice.h, [1579](#)
- MCC
  - nas\_CDMA SysInfo, [184](#)
  - nas\_CDMA SysInfoExt, [185](#)
  - nas\_currentPLMN, [192](#)
  - nas\_ForbiddenNetworks3GPP, [205](#)
  - nas\_GSM SysInfo, [213](#)
  - nas\_lteOpModeTlv, [242](#)
  - nas\_LTE SysInfo, [252](#)
  - nas\_MNCPCSDigitStatus, [256](#)
  - nas\_NR5GSystemInfoTlv, [275](#)
  - nas\_QmiNas3GppNetworkInfo, [292](#)
  - nas\_QmiNas3GppNetworkRAT, [293](#)
  - nas\_QmisNasPcsDigit, [294](#)
  - nas\_WCDMA SysInfo, [351](#)
  - unpack\_nas\_GetServingNetwork\_t, [804](#)
- MDMCallDuration
  - connectionStatus, [53](#)
- MDMConnStatus
  - connectionStatus, [53](#)
  - wds\_ConnStatusTlv, [1159](#)
- MEID\_MAX\_SIZE
  - dms.h, [1207](#)
- MEIDString
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [677](#)
- MIN
  - unpack\_dms\_GetVoiceNumber\_t, [693](#)
- MINREQBKLEN
  - common.h, [1194](#)
- MMTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, [935](#)
- MNC
  - LibPackProfileMnc, [141](#)
  - nas\_CDMA SysInfo, [184](#)
  - nas\_currentPLMN, [192](#)
  - nas\_ForbiddenNetworks3GPP, [205](#)
  - nas\_GSM SysInfo, [213](#)
  - nas\_lteOpModeTlv, [242](#)
  - nas\_LTE SysInfo, [252](#)
  - nas\_MNCPCSDigitStatus, [256](#)
  - nas\_NR5GSystemInfoTlv, [275](#)
  - nas\_QmiNas3GppNetworkInfo, [292](#)
  - nas\_QmiNas3GppNetworkRAT, [293](#)
  - nas\_QmisNasPcsDigit, [294](#)
  - nas\_WCDMA SysInfo, [351](#)
  - unpack\_nas\_GetServingNetwork\_t, [804](#)
- MNCIncPCSDigit
  - nas\_MNCPCSDigitStatus, [256](#)
- MNCPCSDigitStatTlv
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [830](#)
- MPTlv
  - NASQmiCbkNasSystemSelPrefInd, [368](#)
- MSETlv
  - unpack\_ims\_SLQSVolIPCfgCallBack\_ind\_t, [753](#)
- MSGID\_AND\_LEN
  - common.h, [1194](#)
- MSGID\_DONT\_CARE
  - common.h, [1194](#)
- MTMessageInfo



- newMTMessageTlv, 372
- MTUSize
  - pack\_swidms\_SLQSSwiDmsSetMTU\_t, 510
- MTUSize3gpp
  - swidms\_mtuSize3gppTlv, 617
- manString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, 392
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 724
- manufacturer
  - unpack\_dms\_GetManufacturer\_t, 683
- Mask
  - pack\_wds\_SLQSGetDUNCallInfo\_t, 569
- mask
  - loc\_BdsSV, 150
  - loc\_CellDb, 152
  - loc\_ClkInfo, 153
  - loc\_GnssData, 155
  - loc\_SV, 167
  - unpack\_qos\_IPv6TrafCls\_t, 908
  - unpack\_qos\_Tos\_t, 927
- max
  - PackSwiAVMSSettingsPeriodsInfo, 586
- max\_channel\_rx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_t, 1076
- max\_channel\_tx\_rate
  - unpack\_wds\_SLQSGetCurrentChannelRate\_t, 1076
- max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- max\_dist\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- max\_time
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- max\_time\_reported
  - unpack\_swiloc\_SwiLocGetAutoStart\_t, 971
- MaxActive
  - dms\_devMaxCfgListCaps, 59
  - dms\_devMultiSimVoiceDataCaps, 61
  - dms\_devSubsCfgList, 62
- MaxAllowedPktSz
  - unpack\_qos\_swiQosFlow\_t, 926
- MaxChanRxRate
  - dunchannelRate, 77
- MaxChanTxRate
  - dunchannelRate, 77
- maxChannelRXRate
  - unpack\_wds\_GetConnectionRate\_t, 1051
- maxChannelTXRate
  - unpack\_wds\_GetConnectionRate\_t, 1051
- maxDIBitRate
  - LibPackQosClassID, 142
- maxDownlinkBitrate
  - LibPackUMTSQoS, 146
  - wds\_UMTSMInQoS, 1179
- maxImages
  - FMSImageIDEntries, 81
- maxMitigationLevel
  - tmd\_mitigationDevList, 621
- MaxRXChannelRate
  - dms\_devCaps, 57
  - unpack\_dms\_GetDeviceCap\_t, 669
- maxRxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, 671
- maxSDUSize
  - LibPackUMTSQoS, 146
  - wds\_UMTSMInQoS, 1179
- maxStorageSize
  - sms\_maxStorageSizeResp, 600
- MaxSubs
  - dms\_devMaxCfgListCaps, 59
  - dms\_devMultiSimCaps, 60
  - dms\_devMultiSimVoiceDataCaps, 61
- MaxSubsCapLen
  - dms\_devMaxSubsCaps, 60
- MaxSubsList
  - dms\_devMaxSubsCaps, 60
- MaxTXChannelRate
  - dms\_devCaps, 57
  - unpack\_dms\_GetDeviceCap\_t, 669
- maxTxChannelRate
  - unpack\_dms\_GetDeviceCapabilities\_t, 671
- maxUIBitRate
  - LibPackQosClassID, 142
- maxUplinkBitrate
  - LibPackUMTSQoS, 146
  - wds\_UMTSMInQoS, 1179
- mcc
  - nas\_CSGID, 191
  - nas\_MNRInfo, 256
  - nas\_netSelectionPref, 258
  - nas\_NR5GSysInfoTlv, 272
  - nas\_OperatorPLMNData, 278
  - nas\_PlmnID, 286
  - pack\_nas\_SLQSGetPLMNName\_t, 433
  - unpack\_nas\_GetHomeNetwork\_t, 800
  - unpack\_nas\_SLQSGetHomeNetwork\_t, 815
- mccIncPcsDigit
  - nas\_homeNwMNC3GppTlv, 220
- mccM
  - nas\_minBasedIMSI, 255
- mccT
  - nas\_trueIMSI, 335
- mdmCallDurationActive
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1080
- mdn
  - nas\_Mdn, 254
- mdnLen
  - nas\_Mdn, 254
- meanThroughputClass
  - LibPackGPRSRequestedQoS, 105
  - wds\_GPRSQoS, 1169
- meid
  - unpack\_dms\_GetSerialNumbers\_t, 690

- meidSize
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [677](#)
- meidString
  - unpack\_dms\_SLQSSwiGetSerialNoExt\_t, [726](#)
- memoryDumpAllowed
  - unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, [966](#)
- message
  - unpack\_sms\_SLQSGetSMS\_t, [938](#)
- message\_type
  - NASOTAMessageTlv, [360](#)
- messageClass
  - sms\_routeEntry, [603](#)
- messageFailureCode
  - unpack\_sms\_SendSMS\_t, [933](#)
- messageFormat
  - pack\_sms\_SaveSMS\_t, [489](#)
  - pack\_sms\_SendSMS\_t, [489](#)
  - sms\_sendAsynsmsParams, [605](#)
  - unpack\_sms\_SLQSGetSMS\_t, [938](#)
- messageID
  - unpack\_sms\_SendSMS\_t, [933](#)
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t, [949](#)
- messageIndex
  - pack\_sms\_SLQSGetSMS\_t, [492](#)
  - pack\_sms\_SLQSMModifySMSStatus\_t, [494](#)
  - qmiSmsMessageList, [587](#)
  - sMSMTMessageInfo, [611](#)
- messageList
  - unpack\_sms\_SLQSGetSMSList\_t, [940](#)
- messageListSize
  - unpack\_sms\_SLQSGetSMSList\_t, [940](#)
- messageMode
  - sMSMessageModelInfo, [611](#)
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, [950](#)
- MessageModelInfo
  - messageModeTlv, [171](#)
- messageModeTlv, [170](#)
  - MessageModelInfo, [171](#)
  - TlvPresent, [171](#)
- messageSize
  - pack\_sms\_SaveSMS\_t, [489](#)
  - pack\_sms\_SendSMS\_t, [489](#)
  - sms\_sendAsynsmsParams, [605](#)
  - unpack\_sms\_SLQSGetSMS\_t, [938](#)
- messageTag
  - pack\_sms\_SLQSMModifySMSStatus\_t, [494](#)
  - qmiSmsMessageList, [587](#)
  - unpack\_sms\_SLQSGetSMS\_t, [938](#)
- messageType
  - sms\_routeEntry, [603](#)
- MicMute
  - pack\_audio\_SLQSSetAudioProfile\_t, [377](#)
  - pack\_swiaudio\_SLQSSetM2MAVMute\_t, [503](#)
  - unpack\_audio\_SLQSGetAudioProfile\_t, [658](#)
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [951](#)
  - unpack\_swiaudio\_SLQSGetM2MAVMute\_t, [953](#)
- min
  - PackSwiAVMSSettingsPeriodsInfo, [586](#)
- MinPolicedPktSz
  - unpack\_qos\_swiQosFlow\_t, [926](#)
- minSessExp
  - ims\_MinSessExpInfo, [90](#)
- minSize
  - unpack\_dms\_GetVoiceNumber\_t, [693](#)
- minute
  - nas\_timeInfo, [334](#)
  - nas\_UniversalTime, [343](#)
- mipMode
  - unpack\_wds\_GetMobileIP\_t, [1060](#)
- mipStatus
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- mipstatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- mitigationDevID
  - pack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, [519](#)
  - pack\_tmd\_SLQSTmdGetMitigationLvl\_t, [520](#)
  - pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, [520](#)
- mitigationDevIDLen
  - pack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, [519](#)
  - pack\_tmd\_SLQSTmdGetMitigationLvl\_t, [520](#)
  - pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, [520](#)
- mitigationDevId
  - tmd\_mitigationDevList, [621](#)
- mitigationDevIdLen
  - tmd\_mitigationDevList, [621](#)
- MitigationDevList
  - unpack\_tmd\_SLQSTmdGetMitigationDevList\_t, [982](#)
- MitigationDevListLen
  - unpack\_tmd\_SLQSTmdGetMitigationDevList\_t, [982](#)
- mnc
  - nas\_CSGID, [191](#)
  - nas\_MNRInfo, [256](#)
  - nas\_netSelectionPref, [258](#)
  - nas\_NR5GSysInfoTlv, [272](#)
  - nas\_OperatorPLMNData, [278](#)
  - nas\_PlmnID, [286](#)
  - pack\_nas\_SLQSGetPLMNName\_t, [433](#)
  - unpack\_nas\_GetHomeNetwork\_t, [800](#)
  - unpack\_nas\_SLQSGetHomeNetwork\_t, [816](#)
- mncPcsDigits
  - nas\_CSGID, [191](#)
- mobileCountryCode
  - sMSEtwSPlmnInfo, [610](#)
- mobileIP
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [575](#)



- mobileNetworkCode
  - sMSEtwsPlmnInfo, [610](#)
- mode
  - pack\_dms\_SetEventReport\_t, [386](#)
  - pack\_dms\_SetPower\_t, [388](#)
  - pack\_loc\_SetOperationMode\_t, [412](#)
  - pack\_sms\_SLQSSetSmsBroadcastConfig\_t, [492](#)
  - pack\_sms\_SLQSSetSmsBroadcastActivation\_t, [495](#)
  - pack\_sms\_SLQSSetSmsBroadcastConfig\_t, [496](#)
  - pack\_wds\_SetMobileIP\_t, [563](#)
  - uim\_refreshevent, [643](#)
  - voice\_callInfo, [1129](#)
- ModePref
  - nas\_ModePrefTlv, [257](#)
  - NASModePreferenceTlv, [358](#)
- modelIdStr
  - \_litefw\_FirmwareFileInfo, [36](#)
- modelString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, [392](#)
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, [724](#)
- modelid
  - unpack\_dms\_GetModelID\_t, [684](#)
- modelid\_str
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
- modemMode
  - nas\_CommInfo, [190](#)
- modemindex
  - pack\_fms\_SetImagesPreference\_t, [400](#)
- month
  - nas\_timeInfo, [334](#)
  - nas\_UniversalTime, [343](#)
- msgCount
  - sms\_messageWaitingInfoContent, [600](#)
- msgDelFailureCause
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-\_ind\_t, [949](#)
- msgDelFailureType
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-\_ind\_t, [949](#)
- msgProtocol
  - sms\_msgProtocolResp, [601](#)
- msgType
  - sms\_messageWaitingInfoContent, [600](#)
- msgWaitInfo
  - sms\_getMsgWaitingInfo, [597](#)
  - unpack\_sms\_SLQSWmsMessageWaitingCall-Back\_ind\_t, [950](#)
- msgid
  - pack\_qmi\_t, [485](#)
  - unpack\_qmi\_t, [906](#)
- msgtype
  - common.h, [1196](#)
- Mtu
  - unpack\_wds\_SLQSSetRuntimeSettings\_t, [1084](#)
- MultDisc
  - nas\_protocolSubtypeElement, [290](#)
- multiplier
  - unpack\_qos\_pktErrRate\_t, [909](#)
- MuxID
  - qos\_BindDataPortMuxID\_t, [588](#)
- NAI
  - unpack\_wds\_GetMobileIPProfile\_t, [1062](#)
- NAS\_NAM\_NAME\_LENGTH
  - nas.h, [1332](#)
- NAS\_PLMN\_LENGTH
  - nas.h, [1332](#)
- NASAcqOrderPrefTlv, [352](#)
  - AcqOrderLen, [352](#)
  - AcqOrderPref, [352](#)
  - TlvPresent, [352](#)
- NASBandPreferenceTlv, [352](#)
  - band\_pref, [353](#)
  - TlvPresent, [353](#)
- NASCiotAcqOrderPrefTlv, [353](#)
  - CiotAcqOrderLen, [353](#)
  - CiotAcqOrderPref, [353](#)
  - TlvPresent, [353](#)
- NASCiotLteOpModePrefTlv, [353](#)
  - CiotLteOpModePref, [354](#)
  - TlvPresent, [354](#)
- NASEmergencyModeTlv, [354](#)
  - EmerMode, [354](#)
  - TlvPresent, [354](#)
- NASGWAcqOrderPrefTlv, [355](#)
  - GWAcqOrderPref, [356](#)
  - TlvPresent, [356](#)
- NASLTEBandPreferenceTlv, [356](#)
  - LTETBandPref, [356](#)
  - TlvPresent, [356](#)
- NASLteM1BandPrefTlv, [356](#)
  - LteM1BandPref, [356](#)
  - TlvPresent, [357](#)
- NASLteNB1BandPrefTlv, [357](#)
  - LteNB1BandPref, [358](#)
  - TlvPresent, [358](#)
- NASLteNasReleaseInfoTlv, [357](#)
  - nas\_major, [357](#)
  - nas\_minor, [357](#)
  - nas\_release, [357](#)
  - TlvPresent, [357](#)
- NASModePreferenceTlv, [358](#)
  - ModePref, [358](#)
  - TlvPresent, [358](#)
- NASNetSelPreferenceTlv, [358](#)
  - NetSelPref, [358](#)
  - TlvPresent, [358](#)
- NASNr5gBandPrefTlv, [359](#)
  - Nr5gBandPrefbits1\_64, [359](#)
  - Nr5gBandPrefbits\_129\_192, [359](#)
  - Nr5gBandPrefbits\_193\_256, [359](#)
  - Nr5gBandPrefbits\_65\_128, [359](#)
  - TlvPresent, [359](#)
- NASOTAMessageTlv, [359](#)
  - data\_buf, [360](#)
  - data\_len, [360](#)

- message\_type, 360
- TlvPresent, 360
- NASPRLPreferenceTlv, 365
  - PRLPref, 365
  - TlvPresent, 366
- NASPhyCaAggPcellInfo, 360
  - dl\_bw\_value, 361
  - freq, 361
  - iLTEbandValue, 361
  - pci, 361
  - TlvPresent, 361
- NASPhyCaAggScellArray, 361
  - band, 362
  - cphy\_ca\_dl\_bandwidth, 362
  - cphy\_scell\_info\_list\_len, 362
  - freq, 362
  - pci, 362
  - scell\_idx, 362
  - scell\_state, 362
  - TlvPresent, 362
- NASPhyCaAggScellDIBw, 362
  - dl\_bw\_value, 363
  - TlvPresent, 363
- NASPhyCaAggScellIndType, 363
  - freq, 364
  - pci, 364
  - scell\_state, 364
  - TlvPresent, 364
- NASPhyCaAggScellIndex, 363
  - scell\_idx, 363
  - TlvPresent, 363
- NASPhyCaAggScellInfo, 364
  - dl\_bw\_value, 365
  - freq, 365
  - iLTEbandValue, 365
  - pci, 365
  - scell\_state, 365
  - TlvPresent, 365
- NASQmiCbkNasSwiOTAMessageInd, 366
  - nasRelInfoTlv, 366
  - otaMsgTlv, 366
  - timeTlv, 366
- NASQmiCbkNasSystemSelPrefInd, 366
  - AOPTlv, 368
  - BPTlv, 368
  - CiotAOPTlv, 368
  - CiotOpMPTlv, 368
  - EMTlv, 368
  - GWAOPTlv, 368
  - LBPTlv, 368
  - M1BandPTlv, 368
  - MPTlv, 368
  - NB1BandPTlv, 368
  - NR5gBandPTlv, 368
  - NSPTlv, 368
  - PRLPTlv, 368
  - RPTlv, 369
  - RatDMTlv, 369
  - SDPTlv, 369
  - NASRatDisabledMaskTlv, 369
    - RatDisabledMask, 369
    - TlvPresent, 369
  - NASRoamPreferenceTlv, 369
    - RoamPref, 369
    - TlvPresent, 369
  - NASServDomainPrefTlv, 370
    - SrvDomainPref, 370
    - TlvPresent, 370
  - NASServingSystemInfo, 370
    - csAttachState, 371
    - hdrPersonality, 371
    - psAttachState, 371
    - radioInterfaceList, 371
    - radioInterfaceNo, 371
    - registrationState, 371
    - selectedNetwork, 371
  - NASTimeInfoTlv, 371
    - time, 372
    - TlvPresent, 372
- NB1BandPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- NMEADData
  - unpack\_loc\_EventNMEA\_Ind\_t, 770
- NR5gBandPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- NSPTlv
  - NASQmiCbkNasSystemSelPrefInd, 368
- nUsbNetNum
  - pack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t, 511
- NWQoSStatus
  - unpack\_qos\_SLQSQosGetNetworkStatus\_t, 912
- NWRegStat
  - unpack\_sms\_SLQSNWRegInfoCallback\_ind\_t, 942
- naiSize
  - unpack\_wds\_GetMobileIPProfile\_t, 1062
- namID
  - pack\_nas\_SLQSNasGet3GPP2Subscription\_t, 439
  - voice\_airTimer, 1110
  - voice\_prefVoiceSO, 1150
  - voice\_roamTimer, 1154
- namName
  - nas\_namName, 258
- namNameLen
  - nas\_namName, 258
- Name
  - unpack\_nas\_GetServingNetwork\_t, 804
- name
  - loc\_AppProviderInfoTlv, 150
  - unpack\_nas\_GetHomeNetwork\_t, 800
  - unpack\_nas\_SLQSGetHomeNetwork\_t, 816
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- nameLen

- loc\_AppProviderInfoTlv, [150](#)
- voice\_remotePartyName, [1152](#)
- namePI
  - voice\_remotePartyName, [1152](#)
- nameSize
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, [799](#)
  - unpack\_nas\_GetServingNetwork\_t, [805](#)
- nameString
  - pack\_dms\_SLQSSwiSetOSInfo\_t, [392](#)
  - unpack\_dms\_SLQSSwiGetOSInfo\_t, [725](#)
- namelength
  - unpack\_omaDmFotaTlv\_t, [891](#)
- namesize
  - unpack\_wds\_GetDefaultProfile\_t, [1055](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- nas.h
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_1, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_10, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_11, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_12, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_125, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_126, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_127, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_13, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_14, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_17, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_18, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_19, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_2, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_20, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_21, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_23, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_24, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_25, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_250, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_26, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_27, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_28, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_29, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_3, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_30, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_31, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_32, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_33, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_34, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_35, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_36, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_37, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_38, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_39, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_4, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_40, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_41, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_42, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_43, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_46, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_47, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_48, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_5, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_6, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_66, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_7, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_71, [1334](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_8, [1333](#)
  - eLITEQMI\_LTE\_BAND\_CLASS\_E\_UTRA\_OPER-  
ATING\_BAND\_9, [1333](#)
  - eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_100,  
[1334](#)

- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_15, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_25, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_50, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_6, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_CA\_BW\_NRB\_75, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_C-  
ONFIGURED\_ACTIVATED, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_C-  
ONFIGURED\_DEACTIVATED, [1334](#)
- eLITEQMI\_NAS\_LTE\_CPHY\_SCELL\_STATE\_D-  
ECONFIGURED, [1334](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_100, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_15, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_25, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_50, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_6, [1335](#)
- eNAS\_LTE\_CPHY\_CA\_BW\_NRB\_LITE\_75, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGUR-  
ED\_ACTIVATED\_LITE, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_CONFIGUR-  
ED\_DEACTIVATED\_LITE, [1335](#)
- eNAS\_LTE\_CPHY\_SCELL\_STATE\_DECONFIG-  
URED\_LITE, [1335](#)
- LITE\_SYS\_SRV\_DOMAIN\_CAMPED, [1333](#)
- LITE\_SYS\_SRV\_DOMAIN\_CS\_ONLY, [1333](#)
- LITE\_SYS\_SRV\_DOMAIN\_CS\_PS, [1333](#)
- LITE\_SYS\_SRV\_DOMAIN\_NO\_SRV, [1333](#)
- LITE\_SYS\_SRV\_DOMAIN\_PS\_ONLY, [1333](#)
- nas.h, [1319](#)
- NAS\_PLMN\_LENGTH, [1332](#)
- nas\_LTEBandPrefExtTlv, [1332](#)
- nas\_NR5gBandPrefTlv, [1332](#)
- pack\_nas\_GetACCOLC, [1335](#)
- pack\_nas\_GetANAAAAAuthenticationStatus, [1335](#)
- pack\_nas\_GetCDMANetworkParameters, [1336](#)
- pack\_nas\_GetHomeNetwork, [1336](#)
- pack\_nas\_GetHomeNetwork3GPP2, [1336](#)
- pack\_nas\_GetNetworkPreference, [1337](#)
- pack\_nas\_GetRFInfo, [1337](#)
- pack\_nas\_GetServingNetwork, [1337](#)
- pack\_nas\_GetServingNetworkCapabilities, [1337](#)
- pack\_nas\_GetSignalStrengths, [1338](#)
- pack\_nas\_InitiateDomainAttach, [1338](#)
- pack\_nas\_PerformNetworkScan, [1338](#)
- pack\_nas\_PerformNetworkScanPCI, [1339](#)
- pack\_nas\_SLQSConfigSigInfo, [1341](#)
- pack\_nas\_SLQSGetErrorRate, [1341](#)
- pack\_nas\_SLQSGetHomeNetwork, [1342](#)
- pack\_nas\_SLQSGetNetworkTime, [1342](#)
- pack\_nas\_SLQSGetOperatorNameData, [1342](#)
- pack\_nas\_SLQSGetPLMNName, [1343](#)
- pack\_nas\_SLQSGetServingSystem, [1343](#)
- pack\_nas\_SLQSGetServingSystemV2, [1343](#)
- pack\_nas\_SLQSGetSignalStrength, [1344](#)
- pack\_nas\_SLQSGetSysInfo, [1344](#)
- pack\_nas\_SLQSGetSysInfoV2, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPref, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPrefExt, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPrefExtV2, [1346](#)
- pack\_nas\_SLQSInitiateNetworkRegistration, [1346](#)
- pack\_nas\_SLQSNASGetForbiddenNetworks, [1348](#)
- pack\_nas\_SLQSNASGeteDRXParams, [1348](#)
- pack\_nas\_SLQSNASGeteDRXParamsExt, [1348](#)
- pack\_nas\_SLQSNASSeteDRXParams, [1351](#)
- pack\_nas\_SLQSNASSwiGetChannelLock, [1351](#)
- pack\_nas\_SLQSNASSwiSetChannelLock, [1352](#)
- pack\_nas\_SLQSNasConfigSigInfo2, [1346](#)
- pack\_nas\_SLQSNasGet3GPP2Subscription, [1347](#)
- pack\_nas\_SLQSNasGetCellLocationInfo, [1347](#)
- pack\_nas\_SLQSNasGetCellLocationInfoV2, [1347](#)
- pack\_nas\_SLQSNasGetHDRColorCode, [1349](#)
- pack\_nas\_SLQSNasGetRFInfo, [1349](#)
- pack\_nas\_SLQSNasGetSigInfo, [1349](#)
- pack\_nas\_SLQSNasGetTxRxInfo, [1350](#)
- pack\_nas\_SLQSNasIndicationRegisterExt, [1350](#)
- pack\_nas\_SLQSNasIndicationRegisterV2, [1350](#)
- pack\_nas\_SLQSNasSwiIndicationRegister, [1352](#)
- pack\_nas\_SLQSNasSwiModemStatus, [1352](#)
- pack\_nas\_SLQSPerformNetworkScanV2, [1353](#)
- pack\_nas\_SLQSSetBandPreference, [1353](#)
- pack\_nas\_SLQSSetSignalStrengthsCallback, [1353](#)
- pack\_nas\_SLQSSetSysSelectionPref, [1354](#)
- pack\_nas\_SLQSSetSysSelectionPrefExt, [1354](#)
- pack\_nas\_SLQSSwiGetHDRPersonality, [1355](#)
- pack\_nas\_SLQSSwiGetHDRProtSubtype, [1355](#)
- pack\_nas\_SLQSSwiGetHRPDStats, [1355](#)
- pack\_nas\_SLQSSwiGetLteCQI, [1356](#)
- pack\_nas\_SLQSSwiGetLteSccRxInfo, [1356](#)
- pack\_nas\_SLQSSwiNetworkDebug, [1356](#)
- pack\_nas\_SLQSSwiPSDetach, [1357](#)
- pack\_nas\_SetACCOLC, [1339](#)
- pack\_nas\_SetCDMANetworkParameters, [1340](#)
- pack\_nas\_SetLURRejectCallback, [1340](#)
- pack\_nas\_SetNetworkPreference, [1340](#)
- pack\_nas\_SetRFInfoCallback, [1341](#)
- pack\_nas\_SlqsGetLTECphyCAInfo, [1342](#)
- unpack\_nas\_GetACCOLC, [1357](#)
- unpack\_nas\_GetANAAAAAuthenticationStatus, [1357](#)
- unpack\_nas\_GetCDMANetworkParameters, [1358](#)
- unpack\_nas\_GetHomeNetwork, [1358](#)
- unpack\_nas\_GetHomeNetwork3GPP2, [1358](#)
- unpack\_nas\_GetNetworkPreference, [1359](#)
- unpack\_nas\_GetRFInfo, [1359](#)
- unpack\_nas\_GetServingNetwork, [1359](#)
- unpack\_nas\_GetServingNetworkCapabilities, [1360](#)
- unpack\_nas\_GetSignalStrengths, [1360](#)

- unpack\_nas\_InitiateDomainAttach, 1360
- unpack\_nas\_InitiateDomainAttach\_t, 1332
- unpack\_nas\_PerformNetworkScan, 1361
- unpack\_nas\_SLQSCfgSigInfo, 1365
- unpack\_nas\_SLQSCfgSigInfo\_t, 1332
- unpack\_nas\_SLQSGetErrorRate, 1365
- unpack\_nas\_SLQSGetHomeNetwork, 1365
- unpack\_nas\_SLQSGetNetworkTime, 1366
- unpack\_nas\_SLQSGetOperatorNameData, 1366
- unpack\_nas\_SLQSGetPLMNName, 1367
- unpack\_nas\_SLQSGetServingSystem, 1367
- unpack\_nas\_SLQSGetServingSystemV2, 1367
- unpack\_nas\_SLQSGetSignalStrength, 1368
- unpack\_nas\_SLQSGetSysInfo, 1368
- unpack\_nas\_SLQSGetSysInfoV2, 1368
- unpack\_nas\_SLQSGetSysSelectionPref, 1369
- unpack\_nas\_SLQSGetSysSelectionPrefExt, 1369
- unpack\_nas\_SLQSGetSysSelectionPrefExtV2, 1370
- unpack\_nas\_SLQSInitiateNetworkRegistration, 1370
- unpack\_nas\_SLQSInitiateNetworkRegistration\_t, 1332
- unpack\_nas\_SLQSNASGetForbiddenNetworks, 1373
- unpack\_nas\_SLQSNASGetDRXParams, 1372
- unpack\_nas\_SLQSNASGetDRXParamsExt, 1373
- unpack\_nas\_SLQSNASSetDRXParams, 1377
- unpack\_nas\_SLQSNASSetDRXParams\_t, 1332
- unpack\_nas\_SLQSNASSwiGetChannelLock, 1377
- unpack\_nas\_SLQSNASSwiSetChannelLock, 1379
- unpack\_nas\_SLQSNASSwiSetChannelLock\_t, 1332
- unpack\_nas\_SLQSNasConfigSigInfo2, 1370
- unpack\_nas\_SLQSNasConfigSigInfo2\_t, 1332
- unpack\_nas\_SLQSNasEdrxChangeInfoCallBack\_Ind, 1371
- unpack\_nas\_SLQSNasGet3GPP2Subscription, 1371
- unpack\_nas\_SLQSNasGetCellLocationInfo, 1371
- unpack\_nas\_SLQSNasGetCellLocationInfoV2, 1372
- unpack\_nas\_SLQSNasGetHDRColorCode, 1373
- unpack\_nas\_SLQSNasGetRFInfo, 1374
- unpack\_nas\_SLQSNasGetSigInfo, 1374
- unpack\_nas\_SLQSNasGetTxRxInfo, 1374
- unpack\_nas\_SLQSNasIndicationRegisterExt, 1375
- unpack\_nas\_SLQSNasIndicationRegisterExt\_t, 1332
- unpack\_nas\_SLQSNasIndicationRegisterV2, 1375
- unpack\_nas\_SLQSNasIndicationRegisterV2\_t, 1332
- unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind, 1375
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind, 1376
- unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind, 1376
- unpack\_nas\_SLQSNasSigInfoCallback\_ind, 1377
- unpack\_nas\_SLQSNasSwiIndicationRegister, 1378
- unpack\_nas\_SLQSNasSwiIndicationRegister\_t, 1332
- unpack\_nas\_SLQSNasSwiModemStatus, 1378
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind, 1378
- unpack\_nas\_SLQSNasSysInfoCallback\_ind, 1379
- unpack\_nas\_SLQSNasTimerCallback\_ind, 1379
- unpack\_nas\_SLQSPerformNetworkScanV2, 1380
- unpack\_nas\_SLQSSetBandPreference, 1380
- unpack\_nas\_SLQSSetBandPreference\_t, 1333
- unpack\_nas\_SLQSSetSignalStrengthsCallback, 1380
- unpack\_nas\_SLQSSetSignalStrengthsCallback\_t, 1333
- unpack\_nas\_SLQSSetSysSelectionPref, 1381
- unpack\_nas\_SLQSSetSysSelectionPref\_t, 1333
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind, 1381
- unpack\_nas\_SLQSSetSysSelectionPrefExt, 1381
- unpack\_nas\_SLQSSetSysSelectionPrefExt\_t, 1333
- unpack\_nas\_SLQSSwiGetHDRPersonality, 1382
- unpack\_nas\_SLQSSwiGetHDRProtSubtype, 1382
- unpack\_nas\_SLQSSwiGetHRPDStats, 1382
- unpack\_nas\_SLQSSwiGetLteCQI, 1383
- unpack\_nas\_SLQSSwiGetLteSccRxInfo, 1383
- unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind, 1383
- unpack\_nas\_SLQSSwiNetworkDebug, 1384
- unpack\_nas\_SLQSSwiPSDetach, 1384
- unpack\_nas\_SLQSSwiPSDetach\_t, 1333
- unpack\_nas\_SLQSSwiRandIndicatorCallback\_Ind, 1384
- unpack\_nas\_SetACCOLC, 1361
- unpack\_nas\_SetACCOLC\_t, 1332
- unpack\_nas\_SetCDMANetworkParameters, 1361
- unpack\_nas\_SetCDMANetworkParameters\_t, 1332
- unpack\_nas\_SetDataCapabilitiesCallback\_ind, 1362
- unpack\_nas\_SetEventReportInd, 1362
- unpack\_nas\_SetLURejectCallback, 1363
- unpack\_nas\_SetLURejectCallback\_t, 1332
- unpack\_nas\_SetNasLTECphyCalIndCallback\_ind, 1363
- unpack\_nas\_SetNetworkPreference, 1363
- unpack\_nas\_SetRFInfoCallback, 1364
- unpack\_nas\_SetRFInfoCallback\_t, 1332
- unpack\_nas\_SetRoamingIndicatorCallback\_ind, 1364
- unpack\_nas\_SetServingSystemCallback\_ind, 1364
- unpack\_nas\_SlqsGetLTECphyCAInfo, 1366



- unpack\_valid\_nas\_GetCDMANetworkParameters, 1385
- unpack\_valid\_nas\_SLQSGetServingSystem, 1385
- unpack\_valid\_nas\_SLQSGetSignalStrength, 1386
- unpack\_valid\_nas\_SLQSNasGetSigInfo, 1387
- nas\_AcqOrderPrefTlv, 171
  - acqOrdeLen, 172
  - pAcqOrder, 172
  - TlvPresent, 172
- nas\_ActPilotPNElement, 172
  - ActSetPilotPN, 173
  - ActSetPilotPNStrength, 173
- nas\_AddCDMASysInfo, 173
  - geoSysIdx, 173
  - regPrd, 173
- nas\_AddSysInfo, 173
  - cellBroadcastCap, 174
  - geoSysIdx, 174
- nas\_BandPrefInfoTlv, 174
  - bits\_129\_192, 174
  - bits\_193\_256, 174
  - bits\_1\_64, 174
  - bits\_65\_128, 175
  - TlvPresent, 175
- nas\_BandPrefTlv, 175
  - BandPref, 177
  - TlvPresent, 177
- nas\_CDMAChannel, 178
  - priChA, 179
  - priChB, 179
  - secChA, 179
  - secChB, 179
- nas\_CDMAECIOThresh, 179
  - CDMAECIOThreshListLen, 179
  - pCDMAECIOThreshList, 179
- nas\_CDMAInfo, 180
  - baselId, 180
  - baseLat, 180
  - baseLong, 180
  - nid, 180
  - refpn, 181
  - sid, 181
- nas\_CDMARSSIThresh, 181
  - CDMARSSIThreshListLen, 181
  - pCDMARSSIThreshList, 181
- nas\_CDMASysInfo, 181
  - baselId, 184
  - baseLat, 184
  - baseLong, 184
  - bsInfoValid, 184
  - bsPRev, 184
  - bsPRevValid, 184
  - ccsSupported, 184
  - ccsSupportedValid, 184
  - cdmaSysIdValid, 184
  - isSysPrIMatch, 184
  - isSysPrIMatchValid, 184
  - MCC, 184
  - MNC, 184
  - networkID, 184
  - networkIdValid, 184
  - pRevInUse, 185
  - pRevInUseValid, 185
  - packetZone, 185
  - packetZoneValid, 185
  - sysInfoCDMA, 185
  - systemID, 185
- nas\_CDMASysInfoExt, 185
  - imsi\_11\_12, 185
  - MCC, 185
- nas\_CSGID, 190
  - id, 191
  - mcc, 191
  - mnc, 191
  - mncPcsDigits, 191
  - rat, 191
- nas\_CallBarringSysInfo, 177
  - csBarStatus, 177
  - psBarStatus, 177
- nas\_CiotAcqOrderPrefTlv, 187
  - ciotAcqOrderLen, 188
  - pCiotAcqOrder, 188
  - TlvPresent, 188
- nas\_CiotLteOpModePrefTlv, 188
  - ciotLteOpModePref, 188
  - TlvPresent, 188
- nas\_CommInfo, 188
  - imsRegState, 190
  - modemMode, 190
  - psState, 190
  - systemMode, 190
  - temperature, 190
- nas\_CsgId, 190
  - csgId, 190
  - TlvPresent, 190
- nas\_DRCPParams, 197
  - DRCCLover, 198
  - DRCValue, 198
- nas\_DataStatusDetail, 193
  - IPAddress, 194
  - LastErrCode, 194
- nas\_DeviceConfigDetail, 196
  - Chipset, 197
  - HWVersion, 197
  - QLIC, 197
  - Technology, 197
- nas\_ECIOThresh, 199
  - ECIOThreshListLen, 199
  - pECIOThreshList, 199
- nas\_EdrxCiotLteMode, 199
  - lteOpMode, 200
  - TlvPresent, 200
- nas\_EdrxCycleLength, 200
  - cycleLength, 200
  - TlvPresent, 200
- nas\_EdrxEnableType, 200

- edrxEnabled, [201](#)
- TlvPresent, [201](#)
- nas\_EdrxPagingTimeWindow, [201](#)
  - edrxPtw, [201](#)
  - TlvPresent, [201](#)
- nas\_EdrxRatType, [201](#)
  - edrxRatType, [202](#)
  - TlvPresent, [202](#)
- nas\_EmerModeTlv, [202](#)
  - EmerMode, [203](#)
  - TlvPresent, [203](#)
- nas\_ForbiddenNetworks3GPP, [204](#)
  - forbiddenNwInstLen, [205](#)
  - MCC, [205](#)
  - MNC, [205](#)
  - TlvPresent, [205](#)
- nas\_GERANInfo, [205](#)
  - arfcn, [206](#)
  - bsic, [206](#)
  - cellID, [206](#)
  - insNmrCellInfo, [206](#)
  - lac, [206](#)
  - nmrInst, [206](#)
  - plmn, [206](#)
  - rxLev, [206](#)
  - timingAdvance, [206](#)
- nas\_GSMRSSIThresh, [209](#)
  - GSMRSSIThreshListLen, [210](#)
  - pGSMRSSIThreshList, [210](#)
- nas\_GSMSrvStatusInfo, [210](#)
  - isPrefDataPath, [210](#)
  - srvStatus, [210](#)
  - trueSrvStatus, [210](#)
- nas\_GSMSysInfo, [211](#)
  - cellId, [213](#)
  - cellIdValid, [213](#)
  - dtmSupp, [213](#)
  - dtmSuppValid, [213](#)
  - egprsSupp, [213](#)
  - egprsSuppValid, [213](#)
  - lac, [213](#)
  - lacValid, [213](#)
  - MCC, [213](#)
  - MNC, [213](#)
  - networkIdValid, [213](#)
  - regRejectInfoValid, [213](#)
  - rejCause, [213](#)
  - rejectSrvDomain, [213](#)
  - sysInfoGSM, [213](#)
- nas\_GWAcqOrderPrefTlv, [213](#)
  - GWAcqOrderPref, [214](#)
  - TlvPresent, [214](#)
- nas\_HDRECIOThresh, [214](#)
  - HDRECIOThreshListLen, [214](#)
  - pHDRECIOThreshList, [214](#)
- nas\_HDRIOThresh, [214](#)
  - HDRIOThreshListLen, [215](#)
  - pHDRIOThreshList, [215](#)
- nas\_HDRPersonality\_Ind\_Data, [215](#)
  - pCurrentPersonality, [216](#)
  - pPersonalityListLength, [216](#)
  - pProtocolSubtypeElement, [216](#)
- nas\_HDRRSSIThresh, [216](#)
  - HDRRSSIThreshListLen, [216](#)
  - pHRRSSIThreshList, [216](#)
- nas\_HDRSINRThresh, [216](#)
  - HDRSINRThreshListLen, [217](#)
  - pHRSINRThreshList, [217](#)
- nas\_HDRSINRThreshold, [217](#)
  - HDRSINRThreshListLen, [217](#)
  - pHRSINRThreshList, [217](#)
- nas\_HDRSysInfo, [218](#)
  - hdrActiveProt, [219](#)
  - hdrActiveProtValid, [219](#)
  - hdrPersonality, [219](#)
  - hdrPersonalityValid, [219](#)
  - is856SysId, [219](#)
  - is856SysIdValid, [219](#)
  - isSysPrIMatch, [219](#)
  - isSysPrIMatchValid, [219](#)
  - sysInfoHDR, [219](#)
- nas\_IOThresh, [223](#)
  - IOThreshListLen, [223](#)
  - pIOThreshList, [223](#)
- nas\_ImsVoiceSupportLteTlv, [221](#)
  - ImsVoiceSupportLte, [221](#)
  - TLVPresent, [221](#)
- nas\_LTEBandPrefExtTlv
  - nas.h, [1332](#)
- nas\_LTEBandPrefTlv, [224](#)
  - LTEBandPref, [225](#)
  - TlvPresent, [226](#)
- nas\_LTEInfo, [230](#)
  - band, [231](#)
  - bandwidth, [231](#)
  - emmConnState, [232](#)
  - emmState, [232](#)
  - emmSubState, [232](#)
  - RXChan, [232](#)
  - TXChan, [232](#)
- nas\_LTEInfoInterfreq, [232](#)
  - freqsLen, [232](#)
  - InfoInterfreq, [232](#)
  - ueInIdle, [232](#)
- nas\_LTEInfoIntrafreq, [232](#)
  - CellParams, [234](#)
  - cellReselPriority, [234](#)
  - cellsLen, [234](#)
  - earfcn, [234](#)
  - globalCellId, [234](#)
  - plmn, [234](#)
  - sIntraSearch, [234](#)
  - sNonIntraSearch, [234](#)
  - servingCellId, [234](#)
  - tac, [234](#)
  - threshServingLow, [234](#)

- ueInIdle, [234](#)
- nas\_LTEInfoNeighboringGSM, [234](#)
  - freqsLen, [235](#)
  - LteGsmCellInfo, [235](#)
  - ueInIdle, [235](#)
- nas\_LTEInfoNeighboringWCDMA, [235](#)
  - freqsLen, [236](#)
  - ueInIdle, [236](#)
- nas\_LTEOperationMode, [240](#)
  - pLTEOperationMode, [241](#)
  - TlvPresent, [241](#)
- nas\_LTEOperationalModeTlv, [240](#)
  - ciotLteOpMode, [240](#)
  - TlvPresent, [240](#)
- nas\_LTERSRPThresh, [244](#)
  - LTERSRPThreshListLen, [245](#)
  - pLTERSRPThreshList, [245](#)
- nas\_LTERSRQThresh, [245](#)
  - LTERSRQThreshListLen, [245](#)
  - pLTERSRQThreshList, [245](#)
- nas\_LTERSSIThresh, [245](#)
  - LTERSSIThreshListLen, [246](#)
  - pLTERSSIThreshList, [246](#)
- nas\_LTESNRThresh, [248](#)
  - LTESNRThreshListLen, [249](#)
  - pLTESNRThreshList, [249](#)
- nas\_LTESNRThreshold, [249](#)
  - LTESNRThreshListLen, [249](#)
  - pLTESNRThreshList, [249](#)
- nas\_LTESigRptCfg, [246](#)
  - avgPeriod, [247](#)
  - rptRate, [247](#)
- nas\_LTESigRptConfig, [247](#)
  - avgPeriod, [248](#)
  - rptRate, [248](#)
- nas\_LTESysInfo, [250](#)
  - cellId, [251](#)
  - cellIdValid, [252](#)
  - lac, [252](#)
  - lacValid, [252](#)
  - MCC, [252](#)
  - MNC, [252](#)
  - networkIdValid, [252](#)
  - regRejectInfoValid, [252](#)
  - rejCause, [252](#)
  - rejectSrvDomain, [252](#)
  - sysInfoLTE, [252](#)
  - tac, [252](#)
  - tacValid, [252](#)
- nas\_LteCiotOpModeTlv, [226](#)
  - campedCiotLteOpMode, [226](#)
  - TlvPresent, [226](#)
- nas\_LteEarfcnInfo, [227](#)
  - lteEarfcn, [227](#)
  - lteInterEarfcnlen, [227](#)
  - TlvPresent, [227](#)
- nas\_LteEmbmsCoverageTlv, [227](#)
  - LteEmbmsCoverage, [228](#)
  - TLVPresent, [228](#)
- nas\_LteEmbmsTraceIdTlv, [228](#)
  - LteEmbmsTraceId, [228](#)
  - TLVPresent, [228](#)
- nas\_LteM1BandPrefTlv, [236](#)
  - lteM1BandPref, [237](#)
  - TlvPresent, [238](#)
- nas\_LteNb1BandPrefTlv, [238](#)
  - lteNb1BandPref, [239](#)
  - TlvPresent, [240](#)
- nas\_LteOpMode, [241](#)
  - lteOpMode, [241](#)
  - TlvPresent, [241](#)
- nas\_LteRegDomainTlv, [243](#)
  - LteRegDomain, [244](#)
  - TLVPresent, [244](#)
- nas\_LteVoiceDomainTlv, [252](#)
  - LteVoiceDomain, [252](#)
  - TLVPresent, [253](#)
- nas\_MNCPCSDigitStatus, [255](#)
  - MCC, [256](#)
  - MNC, [256](#)
  - MNCIncPCSDigit, [256](#)
  - TLVPresent, [256](#)
- nas\_MNRInfo, [256](#)
  - mcc, [256](#)
  - mnc, [256](#)
  - rat, [256](#)
- nas\_Mdn, [254](#)
  - mdn, [254](#)
  - mdnLen, [254](#)
- nas\_ModePrefTlv, [256](#)
  - ModePref, [257](#)
  - TlvPresent, [257](#)
- nas\_NR5GCellStatusInfoTlv, [265](#)
  - nr5gCellStatus, [266](#)
  - TLVPresent, [266](#)
- nas\_NR5GCellStatusTlv, [266](#)
  - nr5gCellStatus, [266](#)
  - TlvPresent, [266](#)
- nas\_NR5GSerStatTlv, [267](#)
  - isPrefDataPath, [267](#)
  - srvStatus, [267](#)
  - TlvPresent, [267](#)
  - trueSrvStatus, [268](#)
- nas\_NR5GSrvStatusTlv, [268](#)
  - isPrefDataPath, [268](#)
  - srvStatus, [268](#)
  - TLVPresent, [269](#)
  - trueSrvStatus, [269](#)
- nas\_NR5GSysInfoTlv, [269](#)
  - cellId, [271](#)
  - cellIdValid, [271](#)
  - isSysForbidden, [271](#)
  - isSysForbiddenValid, [271](#)
  - lac, [271](#)
  - lacValid, [271](#)
  - mcc, [272](#)



- mnc, [272](#)
- nwldValid, [272](#)
- regRejectInfoValid, [272](#)
- rejectCause, [272](#)
- rejectSrvDomain, [272](#)
- roamStatus, [272](#)
- roamStatusValid, [272](#)
- srvCapability, [272](#)
- srvCapabilityValid, [272](#)
- srvDomain, [272](#)
- srvDomainValid, [272](#)
- TLVPresent, [272](#)
- tac, [272](#)
- tacValid, [272](#)
- nas\_NR5GSystemInfoTlv, [272](#)
  - cellId, [275](#)
  - cellIdValid, [275](#)
  - lac, [275](#)
  - lacValid, [275](#)
  - MCC, [275](#)
  - MNC, [275](#)
  - nwldValid, [275](#)
  - regRejectInfoValid, [275](#)
  - rejCause, [275](#)
  - rejectSrvDomain, [275](#)
  - roamStatus, [275](#)
  - roamStatusValid, [275](#)
  - srvCapValid, [276](#)
  - srvDomain, [276](#)
  - srvDomainValid, [276](#)
  - svrcapability, [276](#)
  - sysForbidden, [276](#)
  - sysForbiddenValid, [276](#)
  - tac, [276](#)
  - tacValid, [276](#)
  - TlvPresent, [276](#)
- nas\_NR5gBandPrefTlv
  - nas.h, [1332](#)
- nas\_NetSelPrefTlv, [258](#)
  - NetSelPref, [259](#)
  - TlvPresent, [259](#)
- nas\_NetworkStat1x, [260](#)
  - ActSetCnt, [261](#)
  - NeighborSetCnt, [261](#)
  - pActPilotPNElements, [261](#)
  - pNeighborSetPilotPN, [261](#)
  - RX\_EC\_IO, [262](#)
  - RX\_PWR, [262](#)
  - SO, [262](#)
  - State, [262](#)
  - TX\_PWR, [262](#)
- nas\_NetworkStatEVDO, [262](#)
  - MACIndex, [263](#)
  - PER, [263](#)
  - pSectorID, [263](#)
  - PilotEnergy, [263](#)
  - RX\_PWR, [263](#)
  - SNR, [263](#)
  - SectorIDLen, [263](#)
  - State, [263](#)
- nas\_NumScellsConfig, [276](#)
  - numScellsCfg, [276](#)
  - TlvPresent, [276](#)
- nas\_OperatorPLMNData, [277](#)
  - lac1, [278](#)
  - lac2, [278](#)
  - mcc, [278](#)
  - mnc, [278](#)
  - PLMNRecID, [278](#)
- nas\_PLMNNetworkName, [286](#)
  - numInstance, [286](#)
  - PLMNNetName, [286](#)
- nas\_PLMNNetworkNameData, [286](#)
  - codingScheme, [288](#)
  - countryInitials, [288](#)
  - longName, [288](#)
  - longNameLen, [288](#)
  - longNameSpareBits, [288](#)
  - shortName, [288](#)
  - shortNameLen, [288](#)
  - shortNameSpareBits, [288](#)
- nas\_PRLPrefTlv, [288](#)
  - PRLPref, [289](#)
  - TlvPresent, [289](#)
- nas\_PhyCaAggDIBW, [279](#)
  - aggDIBW, [279](#)
  - TlvPresent, [279](#)
- nas\_PhyCaAggPcellInfo, [279](#)
  - dl\_bw\_value, [280](#)
  - freq, [280](#)
  - iLTEbandValue, [280](#)
  - pci, [280](#)
  - TlvPresent, [280](#)
- nas\_PhyCaAggScellDIBw, [280](#)
  - dl\_bw\_value, [281](#)
  - TlvPresent, [281](#)
- nas\_PhyCaAggScellIndType, [281](#)
  - freq, [282](#)
  - pci, [282](#)
  - scell\_state, [282](#)
  - TlvPresent, [282](#)
- nas\_PhyCaAggScellIndex, [281](#)
  - scell\_idx, [281](#)
  - TlvPresent, [281](#)
- nas\_PhyCaAggScellInfo, [282](#)
  - dl\_bw\_value, [284](#)
  - freq, [284](#)
  - iLTEbandValue, [284](#)
  - pci, [284](#)
  - scell\_state, [284](#)
  - TlvPresent, [284](#)
- nas\_PilotSetData, [284](#)
  - NumPilots, [284](#)
  - pPilotSetInfo, [284](#)
- nas\_PilotSetParams, [285](#)
  - PilotPN, [285](#)

- PilotStrength, 285
- PilotType, 285
- nas\_PlmnID, 285
  - mcc, 286
  - mnc, 286
  - pcsDigit, 286
  - TlvPresent, 286
- nas\_QmiNas3GppNetworkInfo, 291
  - Description, 292
  - Forbidden, 292
  - InUse, 292
  - MCC, 292
  - MNC, 292
  - Preferred, 292
  - Roaming, 292
- nas\_QmiNas3GppNetworkRAT, 292
  - MCC, 293
  - MNC, 293
  - RAT, 293
- nas\_QmisNasPcsDigit, 293
  - includes\_pcs\_digit, 294
  - MCC, 294
  - MNC, 294
- nas\_QmisNasSlqsNasPCICellInfo, 294
  - cellID, 294
  - freq, 294
  - GlobalCellID, 294
  - nasQmisNasPcsDigit, 294
  - PlmnLen, 294
- nas\_QmisNasSlqsNasPCIInfo, 295
  - nasQmisNasSlqsNasPCICellInfo, 295
  - PCICellInfoLen, 295
  - rsrp, 295
  - rsrpRx0, 295
  - rsrpRx1, 295
  - rsrq, 296
  - rsrqRx0, 296
  - rsrqRx1, 296
- nas\_RFBandInfoElements, 297
  - activeBandClass, 298
  - activeChannel, 298
  - radioInterface, 298
- nas\_RFBandInfoExtTlv, 300
  - activeBand, 300
  - activeChannel, 301
  - radioIlf, 301
  - TlvPresent, 301
- nas\_RFBandInfoTlv, 301
  - activeBand, 302
  - activeChannel, 302
  - radioIlf, 302
  - TlvPresent, 302
- nas\_RFBandwidthInfoTlv, 304
  - bandwidth, 305
  - radioIlf, 305
  - TlvPresent, 305
- nas\_RFDedicatedBandInfoTlv, 307
  - dedicatedBand, 307
  - radioIlf, 307
  - TlvPresent, 308
- nas\_RFInfoTlv, 308
  - activeBandClass, 308
  - activeChannel, 308
  - radioInterface, 308
  - radioInterfaceSize, 309
  - TlvPresent, 309
- nas\_RSRPThresh, 310
  - pRSRPThresList, 311
  - RSRPThresListLen, 311
- nas\_RSRQThresh, 311
  - pRSRQThresList, 312
  - RSRQThresListLen, 312
- nas\_RSSIThresh, 312
  - pRSSIThresList, 313
  - RSSIThresListLen, 313
- nas\_RankIndicatorTlv, 296
  - count0, 296
  - count1, 296
  - TlvPresent, 296
- nas\_RatDisabledMaskTlv, 296
  - ratDisabledMask, 297
  - TlvPresent, 297
- nas\_RejectReasonTlv, 297
  - rejectCause, 297
  - serviceDomain, 297
  - TlvPresent, 297
- nas\_RfBandInfoExtFormat, 298
  - pInstancesSize, 299
  - pRfBandInfoExtFormatParam, 299
  - TlvPresent, 299
- nas\_RfBandInfoExtFormatElements, 299
  - activeBand, 299
  - activeChannel, 299
  - radioInterface, 299
- nas\_RfBandInfoList, 301
  - pInstanceSize, 301
  - pRFBandInfoParam, 301
- nas\_RfBandwidthInfo, 302
  - pInstancesSize, 303
  - pRfBandwidthInfoParam, 303
  - TlvPresent, 303
- nas\_RfBandwidthInfoElements, 303
  - bandwidth, 304
  - radioInterface, 304
- nas\_RfDedicatedBandInfo, 305
  - pInstancesSize, 306
  - pRfDedicatedBandInfoParam, 306
  - TlvPresent, 306
- nas\_RfDedicatedBandInfoElements, 306
  - dedicatedBand, 307
  - radioInterface, 307
- nas\_RoamPrefTlv, 310
  - RoamPref, 310
  - TlvPresent, 310
- nas\_RxSigInfo, 314
  - isRadioTuned, 315

- rsrp, 315
- rxChainIndex, 315
- rxPower, 315
- nas\_SLQSSignalStrengthsIndReq, 321
  - ecioDelta, 322
  - ecioThresholdList, 322
  - ecioThresholdListLen, 322
  - ioDelta, 322
  - lteRsrpDelta, 322
  - lteSnrDelta, 322
  - rsrqDelta, 322
  - rxSignalStrengthDelta, 322
  - sinrDelta, 322
  - sinrThresholdList, 322
  - sinrThresholdListLen, 322
- nas\_SLQSSignalStrengthsInformation, 322
  - ecioInfo, 323
  - errorRateInfo, 323
  - io, 324
  - lteRsrpinfo, 324
  - lteSnrinfo, 324
  - rsrqInfo, 324
  - rxSignalStrengthInfo, 324
  - sinr, 324
- nas\_SLQSSignalStrengthsTlv, 324
  - sSLQSSignalStrengthsInfo, 324
  - TlvPresent, 324
- nas\_SccRxInfo, 316
  - numInstances, 317
  - rsrq, 317
  - sigInfo, 317
  - snr, 317
  - TlvPresent, 317
- nas\_SignalStrengthTlv, 320
  - radioInterface, 320
  - signalStrength, 320
  - TlvPresent, 320
- nas\_SimRejInfoTlv, 320
  - SimRejInfo, 321
  - TLVPresent, 321
- nas\_SrvDomainPrefTlv, 324
  - SrvDomainPref, 325
  - TlvPresent, 325
- nas\_SrvRegRestrictTlv, 326
  - SrvRegRestriction, 326
  - TlvPresent, 326
- nas\_SrvRegRestrictionTlv, 325
  - SrvRegRestriction, 325
  - TLVPresent, 325
- nas\_SrvStatusInfo, 326
  - isPrefDataPath, 327
  - srvStatus, 327
- nas\_TDSCDMABandPrefTlv, 329
  - TdscdmaBandPref, 330
  - TlvPresent, 330
- nas\_TDSCDMAECIOThresh, 330
- nas\_TDSCDMARSCPTThresh, 330
- nas\_TDSCDMARSSIThresh, 331
- nas\_TDSCDMASINRCONFTThresh, 331
- nas\_TDSCDMASINRThresh, 332
- nas\_UMTSExtInfo, 335
  - cellId, 337
  - ecio, 337
  - geranInst, 337
  - geranInstElement, 337
  - lac, 337
  - plmn, 337
  - psc, 337
  - rscp, 337
  - squal, 337
  - srxLvl, 337
  - TlvPresent, 337
  - uarfcn, 337
  - umtsInst, 337
  - umtsInstElement, 337
- nas\_UMTSInfo, 337
  - cellID, 339
  - ecio, 339
  - geranInst, 339
  - GeranInstInfo, 339
  - lac, 339
  - plmn, 339
  - psc, 339
  - rscp, 339
  - UMTSInstInfo, 339
  - uarfcn, 339
  - umtsInst, 339
- nas\_UMTSinstInfo, 340
  - umtsEcio, 341
  - umtsPsc, 341
  - umtsRscp, 341
  - umtsUarfcn, 341
- nas\_UniversalTime, 342
  - day, 343
  - dayOfWeek, 343
  - hour, 343
  - minute, 343
  - month, 343
  - second, 343
  - year, 343
- nas\_UsageSettingTlv, 343
  - TlvPresent, 344
  - UsageSetting, 344
- nas\_VoiceDomainPrefTlv, 344
  - TlvPresent, 344
  - VoiceDomainPref, 344
- nas\_WCDMACellInfoExt, 345
  - TlvPresent, 346
  - wAgc, 346
  - wDIBler, 346
  - wTxAgc, 346
- nas\_WCDMAECIOThresh, 346
- nas\_WCDMAInfoLTENeighborCell, 347
  - umtsLTENbrCellLen, 347
  - wcdmaRRCCState, 347
- nas\_WCDMARSSIThresh, 347

- nas\_WCDMA SysInfo, 348
  - cellId, 351
  - cellIdValid, 351
  - hsCallStatus, 351
  - hsCallStatusValid, 351
  - hsInd, 351
  - hsIndValid, 351
  - lac, 351
  - lacValid, 351
  - MCC, 351
  - MNC, 351
  - networkIdValid, 351
  - psc, 351
  - pscValid, 351
  - regRejectInfoValid, 351
  - rejCause, 351
  - rejectSrvDomain, 351
  - sysInfoWCDMA, 351
- nas\_acqOrderPref, 171
  - acqOrdeLen, 171
  - pAcqOrder, 171
- nas\_callBarStatus, 177
  - csBarStatus, 178
  - psBarStatus, 178
- nas\_cellParams, 185
  - pci, 186
  - rsrp, 186
  - rsrq, 186
  - rssi, 186
  - srxlev, 186
- nas\_ciotAcqOrderPref, 186
  - ciotAcqOrderLen, 187
  - pCiotAcqOrder, 187
- nas\_currentPLMN, 191
  - MCC, 192
  - MNC, 192
  - netDescr, 192
  - netDescrLength, 192
- nas\_dataSrvCapabilities, 192
  - dataCapabilities, 193
  - dataCapabilitiesLen, 193
- nas\_detailSvcInfo, 195
  - hdrHybrid, 196
  - hdrSrvStatus, 196
  - isSysForbidden, 196
  - srvCapability, 196
  - srvStatus, 196
- nas\_dirNum, 197
  - dirNum, 197
  - dirNumLen, 197
- nas\_ecioListElement, 198
  - ecio, 198
  - radiolf, 199
- nas\_errorRateListElement, 203
  - errorRate, 204
  - radiolf, 204
- nas\_geranInstArr, 206
  - geranArfcn, 207
  - geranBsicBcc, 207
  - geranBsicNcc, 207
  - geranRank, 207
  - geranRssi, 207
- nas\_geranInstInfo, 207
  - geranArfcn, 208
  - geranBsicBcc, 208
  - geranBsicNcc, 208
  - geranRssi, 208
- nas\_gsmCellInfo, 208
  - arfcn, 209
  - band1900, 209
  - bsicId, 209
  - cellIdValid, 209
  - rssi, 209
  - srxlev, 209
- nas\_homeNwMNC3GppTlv, 220
  - is3GppNw, 220
  - mccIncPcsDigit, 220
  - TlvPresent, 220
- nas\_homeSIDNID, 220
  - numInstances, 221
  - SidNid, 221
- nas\_infoInterFreq, 221
  - cell\_resele\_priority, 222
  - cellInterFreqParams, 222
  - cells\_len, 222
  - earfcn, 222
  - threshXHigh, 222
  - threshXLow, 223
- nas\_lteBandPrefExt, 223
  - bits\_129\_192, 224
  - bits\_193\_256, 224
  - bits\_1\_64, 224
  - bits\_65\_128, 224
- nas\_lteEARFCN, 226
  - earfcn0, 227
  - earfcn1, 227
  - status, 227
- nas\_lteGsmCellInfo, 229
  - cellReselePriority, 229
  - cells\_len, 229
  - GsmCellInfo, 230
  - nccPermitted, 230
  - threshGsmHigh, 230
  - threshGsmLow, 230
- nas\_lteOpModeTlv, 241
  - lteOpMode, 242
  - lteOpModeLen, 242
  - MCC, 242
  - MNC, 242
  - TlvPresent, 242
- nas\_ltePCI, 242
  - earfcn, 243
  - pci, 243
  - status, 243
- nas\_lteRsrpInformation, 244
  - rsrpLevel, 244

- nas\_lteSnrinformation, 248
  - snrlevel, 248
- nas\_lteWcdmaCellInfo, 253
  - cellReselPriority, 253
  - cellsLen, 253
  - threshXhigh, 253
  - threshXlow, 254
  - uarfcn, 254
  - WCDMACellInfo, 254
- nas\_major
  - NASLteNasReleaseInfoTlv, 357
- nas\_minBasedIMSI, 254
  - imsiM1112, 255
  - imsiMS1, 255
  - imsiMS2, 255
  - mccM, 255
- nas\_minor
  - NASLteNasReleaseInfoTlv, 357
- nas\_namName, 257
  - namName, 258
  - namNameLen, 258
- nas\_netSelectionPref, 258
  - mcc, 258
  - mnc, 258
  - netReg, 258
- nas\_networkNameSrcTlv, 259
  - nwNameSrc, 259
  - nwNameSrcLen, 260
  - TlvPresent, 260
- nas\_nmrCellInfo, 263
  - nmrArfcn, 264
  - nmrBsic, 264
  - nmrCellID, 264
  - nmrLac, 264
  - nmrPlmn, 264
  - nmrRxLev, 265
- nas\_nr5gBandPref, 265
  - bits\_129\_192, 265
  - bits\_193\_256, 265
  - bits\_1\_64, 265
  - bits\_65\_128, 265
- nas\_nwNameSrc3GppTlv, 276
  - NwNameSrc3Gpp, 277
  - TlvPresent, 277
- nas\_operatorNameString, 277
  - PLMNName, 277
- nas\_operatorPLMNList, 278
  - numInstance, 279
  - PLMNData, 279
- nas\_protocolSubtypeElement, 289
  - AccessMac, 290
  - AuthProt, 290
  - ControlMac, 290
  - EncryptProt, 290
  - ForwardMac, 290
  - IdleState, 290
  - KeyExchange, 290
  - MultDisc, 290
  - PhysicalLayer, 290
  - ReverseMac, 290
  - SecProt, 290
  - VirtStream, 290
- nas\_qaQmi3Gpp2TimeZone, 290
  - daylightSavings, 291
  - leapSeconds, 291
  - localTimeOffset, 291
- nas\_release
  - NASLteNasReleaseInfoTlv, 357
- nas\_roamIndList, 309
  - numInstances, 309
  - radiolInterface, 309
  - roamIndicator, 309
- nas\_rsrqInformation, 311
  - radiolf, 311
  - rsrq, 311
- nas\_rxInfo, 313
  - ecio, 314
  - isRadioTuned, 314
  - phase, 314
  - rscp, 314
  - rsrp, 314
  - rxPower, 314
- nas\_rxSignalStrengthListElement, 315
  - radiolf, 316
  - rxSignalStrength, 316
- nas\_servSystem, 318
  - csAttachState, 319
  - numRadioInterfaces, 319
  - psAttachState, 319
  - radiolInterface, 319
  - regState, 319
  - selNetwork, 319
- nas\_serviceProviderName, 317
  - displayCondition, 317
  - spn, 317
  - spnLength, 317
- nas\_sidNid, 319
  - nid, 319
  - sid, 319
- nas\_sysInfoCommon, 327
  - isSysForbidden, 329
  - isSysForbiddenValid, 329
  - roamStatus, 329
  - roamStatusValid, 329
  - srvCapability, 329
  - srvCapabilityValid, 329
  - srvDomain, 329
  - srvDomainValid, 329
- nas\_timeInfo, 332
  - day, 334
  - dayLtSavingAdj, 334
  - dayOfWeek, 334
  - hour, 334
  - minute, 334
  - month, 334
  - radiolInterface, 334

- second, 334
- timeZone, 334
- TlvPresent, 334
- year, 334
- nas\_trueIMSI, 334
  - imsiT1112, 335
  - imsiTS1, 335
  - imsiTS2, 335
  - imsiTaddrNum, 335
  - mccT, 335
- nas\_txInfo, 335
  - isInTraffic, 335
  - txPower, 335
- nas\_umtsInstArr, 339
  - umtsEcio, 340
  - umtsPsc, 340
  - umtsRank, 340
  - umtsRscp, 340
  - umtsSet, 340
  - umtsSqual, 340
  - umtsSrxLvl, 340
  - umtsUarfcn, 340
- nas\_umtsLTENbrCell, 341
  - cellsTDD, 342
  - earfcn, 342
  - pci, 342
  - rsrp, 342
  - rsrq, 342
  - srxlev, 342
- nas\_wcdmaCellInfo, 344
  - cpich\_ecno, 345
  - cpich\_rscp, 345
  - psc, 345
  - srxlev, 345
- nas\_wcdmaUARFCN, 351
  - status, 352
  - uarfcn, 352
- NasGetLTECphyCaInfo, 354
  - PhyCaAggPcellInfo, 355
  - PhyCaAggScellArray, 355
  - PhyCaAggScellIDBw, 355
  - PhyCaAggScellIndType, 355
  - PhyCaAggScellIndex, 355
  - PhyCaAggScellInfo, 355
- nasQmisNasPcsDigit
  - nas\_QmisNasSlqsNasPCICellInfo, 294
- nasQmisNasSlqsNasPCICellInfo
  - nas\_QmisNasSlqsNasPCIInfo, 295
- nasRelInfoTlv
  - NASQmiCbkNasSwiOTAMessageInd, 366
- nccPermitted
  - nas\_lteGsmCellInfo, 230
- NeighborSetCnt
  - nas\_NetworkStat1x, 261
- netDescr
  - nas\_currentPLMN, 192
- netDescrLength
  - nas\_currentPLMN, 192
- netInfoLen
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1093
- netReg
  - nas\_netSelectionPref, 258
- NetSelPref
  - nas\_NetSelPrefTlv, 259
  - NASNetSelPreferenceTlv, 358
- NetworkID
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 830
- networkID
  - nas\_CDMA SysInfo, 184
- networkIdValid
  - nas\_CDMA SysInfo, 184
  - nas\_GSM SysInfo, 213
  - nas\_LTE SysInfo, 252
  - nas\_WCDMA SysInfo, 351
- networkInfo
  - voice\_USSDNotificationNetworkInfo, 1156
- networkInfoLen
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 1074
- NetworkType
  - currNetworkInfo, 56
  - wds\_currNetworkInfo, 1160
- NewIMSRegistration
  - unpack\_imsa\_SLQSImsaRegStatusCallBack\_ind\_t, 758
- NewImsRegStatus
  - unpack\_imsa\_SLQSGetIMSARegStatus\_t, 754
- NewMMTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 935
- newMTMessageTlv, 372
  - MTMessageInfo, 372
  - TlvPresent, 372
- newPINLen
  - uim\_unblockUIMPIN, 652
- newPINVal
  - uim\_unblockUIMPIN, 652
- newPasswd
  - pack\_voice\_SLQSVoiceSetCallBarringPassword\_t, 550
- newPasswdAgain
  - pack\_voice\_SLQSVoiceSetCallBarringPassword\_t, 550
- newPin
  - pack\_dms\_UIMUnblockPIN\_t, 398
- newPwd
  - voice\_newPwdData, 1146
- newPwdAgain
  - voice\_newPwdData, 1146
- newValue
  - pack\_dms\_UIMChangePIN\_t, 394
- nextHeader
  - LibPackTFTIDParams, 144
- nid
  - nas\_CDMAInfo, 180

- nas\_sidNid, [319](#)
- unpack\_nas\_GetHomeNetwork\_t, [800](#)
- unpack\_nas\_SLQSSetHomeNetwork\_t, [816](#)
- nmrArfcn
  - nas\_nmrCellInfo, [264](#)
- nmrBsic
  - nas\_nmrCellInfo, [264](#)
- nmrCellID
  - nas\_nmrCellInfo, [264](#)
- nmrInst
  - nas\_GERANInfo, [206](#)
- nmrLac
  - nas\_nmrCellInfo, [264](#)
- nmrPlmn
  - nas\_nmrCellInfo, [264](#)
- nmrRxLev
  - nas\_nmrCellInfo, [265](#)
- noReplyTimer
  - voice\_callFWExtInfo, [1126](#)
  - voice\_callFWInfo, [1127](#)
- notifType
  - unpack\_voice\_SUPSNotificationCallback\_ind\_t, [1044](#)
- notification
  - unpack\_omaDmNotificationsTlv\_t, [892](#)
- notification\_Type
  - unpack\_voice\_USSDNotificationCallback\_ind\_t, [1044](#)
- notificationType
  - sMSEtwsMessageInfo, [609](#)
- notused
  - unpack\_dms\_SetCrashAction\_t, [696](#)
- Nr5gBandPrefbits1\_64
  - NASNr5gBandPrefTlv, [359](#)
- Nr5gBandPrefbits\_129\_192
  - NASNr5gBandPrefTlv, [359](#)
- Nr5gBandPrefbits\_193\_256
  - NASNr5gBandPrefTlv, [359](#)
- Nr5gBandPrefbits\_65\_128
  - NASNr5gBandPrefTlv, [359](#)
- nr5gCellStatus
  - nas\_NR5GCellStatusInfoTlv, [266](#)
  - nas\_NR5GCellStatusTlv, [266](#)
- num\_instances
  - DMScustSettingList, [75](#)
  - sms\_qaQmi3GPP2BroadcastCfgInfo, [602](#)
  - sms\_qaQmi3GPPBroadcastCfgInfo, [602](#)
- numApp
  - slotInf, [594](#)
  - uim\_physlotInfo, [639](#)
  - uim\_slotInfo, [649](#)
- numCrashes
  - crashInformation, [55](#)
- numEntries
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, [721](#)
- numFeatures
  - uim\_personalizationStatus, [638](#)
- numFiles
  - uim\_registerRefresh, [644](#)
- NumFlows
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, [916](#)
- numInstance
  - nas\_operatorPLMNList, [279](#)
  - nas\_PLMNNetworkName, [286](#)
- numInstances
  - nas\_homeSIDNID, [221](#)
  - nas\_roamIndList, [309](#)
  - nas\_SccRxInfo, [317](#)
  - sms\_getMsgWaitingInfo, [597](#)
  - unpack\_sms\_SLQSWmsMessageWaitingCall-Back\_ind\_t, [950](#)
  - voice\_arrAlertingPattern, [1113](#)
  - voice\_arrAlertingType, [1114](#)
  - voice\_arrAlphaID, [1114](#)
  - voice\_arrCalledPartyNum, [1115](#)
  - voice\_arrCallEndReason, [1116](#)
  - voice\_arrCallInfo, [1116](#)
  - voice\_arrConnectPartyNum, [1117](#)
  - voice\_arrDiagInfo, [1117](#)
  - voice\_arrRedirPartyNum, [1118](#)
  - voice\_arrRemotePartyName, [1118](#)
  - voice\_arrRemotePartyNum, [1119](#)
  - voice\_arrSvcOption, [1119](#)
  - voice\_arrUUSInfo, [1120](#)
  - voice\_getCallFWExtInfo, [1144](#)
  - voice\_getCallFWInfo, [1144](#)
  - wds\_DomainNameList, [1168](#)
  - wds\_PCSCFFQDNAddressList, [1172](#)
  - wds\_PCSCFIPv4ServerAddressList, [1173](#)
- numLen
  - voice\_calledPartyInfo, [1122](#)
  - voice\_callFWExtInfo, [1126](#)
  - voice\_callFWInfo, [1128](#)
  - voice\_callingPartyInfo, [1131](#)
  - voice\_peerNumberInfo, [1148](#)
  - voice\_redirNumInfo, [1151](#)
  - voice\_remotePartyNum, [1153](#)
- numOfFiles
  - uim\_refreshevent, [643](#)
- numOfRoutes
  - sms\_setRoutesReq, [607](#)
- numOpt
  - wds\_DHCPLeaseOptTlv, [1162](#)
  - wds\_DHCPv4OptionList, [1166](#)
  - wdsDhcpv4OptionList, [1181](#)
- numPI
  - voice\_peerNumberInfo, [1148](#)
- NumPilots
  - nas\_PilotSetData, [284](#)
- numPlan
  - voice\_calledPartyInfo, [1122](#)
  - voice\_callFWExtInfo, [1127](#)
  - voice\_callingPartyInfo, [1131](#)
  - voice\_connectNumInfo, [1136](#)
  - voice\_peerNumberInfo, [1148](#)



- voice\_redirNumInfo, 1151
- numPresInd
  - voice\_connectNumInfo, 1136
- numQosFlow
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, 915
- numRadioInterfaces
  - nas\_servSystem, 319
- NumRxFilters
  - unpack\_qos\_QosFlowInfo\_t, 910
- numSI
  - voice\_peerNumberInfo, 1148
- numScellsCfg
  - nas\_NumScellsConfig, 276
- numSlot
  - uim\_cardStatus, 629
- NumSupUSBComps
  - unpack\_dms\_GetUSBComp\_t, 692
- NumTxFilters
  - unpack\_qos\_QosFlowInfo\_t, 911
- numType
  - voice\_calledPartyInfo, 1122
  - voice\_callFWExtInfo, 1127
  - voice\_callingPartyInfo, 1131
  - voice\_connectNumInfo, 1137
  - voice\_peerNumberInfo, 1148
  - voice\_redirNumInfo, 1151
- number
  - voice\_calledPartyInfo, 1122
  - voice\_callFWExtInfo, 1126
  - voice\_callFWInfo, 1127
  - voice\_callingPartyInfo, 1131
  - voice\_ECTNum, 1141
  - voice\_peerNumberInfo, 1148
  - voice\_redirNumInfo, 1151
- NumberOfPhySlot
  - uim\_GetSlotsStatusTlv, 636
- NumberOfPhySlotInfo
  - uim\_GetSlotsInfoTlv, 636
- numberPlan
  - voice\_callFwdTypeAndPlan, 1125
- numberType
  - voice\_callFwdTypeAndPlan, 1125
- nwldValid
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 275
- nwNameSrc
  - nas\_networkNameSrcTlv, 259
- NwNameSrc3Gpp
  - nas\_nwNameSrc3GppTlv, 277
- nwNameSrcLen
  - nas\_networkNameSrcTlv, 260
- NxtHdrProto
  - unpack\_qos\_swiQosFilter\_t, 922
- OKtoRefresh
  - pack\_uim\_SLQSUIRefreshOK\_t, 530
- OMADMEEnable
  - pack\_swioma\_SLQSOMADMSetSettingsExt\_t, 518
- OMADMEEnabled
  - unpack\_swiavms\_SLQSAVMSGetSettings\_t, 958
  - unpack\_swiavms\_SLQSAVMSGetSettings\_v2\_t, 960
  - unpack\_swioma\_SLQSOMADMGetSettings\_t, 980
- OMADMLogEnable
  - pack\_swioma\_SLQSOMADMSetSettingsExt\_t, 518
- OTASPStatus
  - unpack\_voice\_OTASPStatusCallback\_ind\_t, 1006
- oddEvenInd
  - voice\_calledPartySubAdd, 1123
- OfflineReason
  - unpack\_dms\_GetPower\_t, 688
- offset
  - uim\_readTransparentInfo, 642
  - uim\_writeTransparentInfo, 655
- oldPINLen
  - uim\_changeUIMPIN, 629
- oldPINVal
  - uim\_changeUIMPIN, 630
- oldPasswd
  - pack\_voice\_SLQSVoiceSetCallBarringPassword\_t, 550
- oldValue
  - pack\_dms\_UIMChangePIN\_t, 394
- opMode
  - unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 703
- opaqueId
  - pack\_loc\_SLQSLOCInjectSensorData\_t, 420
- operatingMode
  - dms\_OperatingModeTlv, 66
- OperatingModeTlv
  - unpack\_dms\_SetEventReport\_ind\_t, 697
- operation
  - pack\_pds\_SetPDSDDefaults\_t, 476
  - uim\_depersonalizationInformation, 631
- OperationMode
  - unpack\_dms\_GetPower\_t, 688
- optCode
  - wds\_DHCPOpt, 1164
  - wds\_DHCPv4Option, 1166
  - wdsDhcpv4Option, 1180
- optList
  - wds\_DHCPLeaseOptTlv, 1162
- optListData
  - wds\_DHCPLeaseOptTlv, 1163
- optVal
  - wds\_DHCPv4Option, 1166
  - wdsDhcpv4Option, 1181
- optValLen
  - wds\_DHCPOpt, 1164
  - wds\_DHCPv4Option, 1166
  - wdsDhcpv4Option, 1181
- otaMsgTlv
  - NASQmiCbkNasSwiOTAMessageInd, 366
- p1Status
  - unpack\_dms\_UIMGetPINStatus\_t, 733



- p1UnblockRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, [733](#)
- p1VerifyRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, [733](#)
- p2Status
  - unpack\_dms\_UIMGetPINStatus\_t, [733](#)
- p2UnblockRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, [733](#)
- p2VerifyRetriesLeft
  - unpack\_dms\_UIMGetPINStatus\_t, [733](#)
- p3GPP2TimeInfo
  - unpack\_nas\_SLQSGetNetworkTime\_t, [817](#)
- p3GPPTimeInfo
  - unpack\_nas\_SLQSGetNetworkTime\_t, [817](#)
- p3GppNetworkInfoInstances
  - unpack\_nas\_PerformNetworkScan\_t, [808](#)
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, [877](#)
- p3GppNetworkInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, [808](#)
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, [877](#)
- p3gppRelease
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [573](#)
- pAAASPI
  - pack\_wds\_SetMobileIPProfile\_t, [566](#)
- PACK\_WDS\_IPV4
  - wds.h, [1615](#)
- PACK\_WDS\_IPV6
  - wds.h, [1615](#)
- pAMRStatus
  - pack\_voice\_SLQSVoiceGetConfig\_t, [545](#)
- pAPN
  - PackSwiAVMSSettingsAPNInfo, [585](#)
- pAPNBearer
  - LibpackProfile3GPPV2, [127](#)
- pAPNClass
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pAPNClass3GPP2
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [139](#)
- pAPNDisabledFlag
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pAPNEnabled3GPP2
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [139](#)
- pAPNInfo
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, [506](#)
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, [508](#)
  - unpack\_swiaavms\_SLQSAVMSGetSettings\_t, [958](#)
  - unpack\_swiaavms\_SLQSAVMSGetSettings\_v2\_t, [960](#)
- pAPNName
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pAPNnameSize
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pAVRXAVCHearroom
  - audio\_RXAVCList, [43](#)
- pAVRXAVCSens
  - audio\_RXAVCList, [43](#)
- pAccelAcceptReady
  - unpack\_loc\_SensorStreamingCallback\_Ind\_t, [787](#)
- pAccelSamplesAccepted
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, [778](#)
- pAccelTempAcceptReady
  - unpack\_loc\_SensorStreamingCallback\_Ind\_t, [787](#)
- pAccelTempSamplesAccepted
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, [778](#)
- pAccessClassBarringInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pAccolc
  - unpack\_nas\_GetACCOLC\_t, [794](#)
- pAccuracy
  - unpack\_pds\_GetPDSDefaults\_t, [893](#)
- pAcqOrder
  - nas\_acqOrderPref, [171](#)
  - nas\_AcqOrderPrefTlv, [172](#)
- pAcqOrderPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [849](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pActPilotPNElements
  - nas\_NetworkStat1x, [261](#)
- pActiveTimer
  - pack\_dms\_SLQSSetPowerSaveModeConfig\_t, [390](#)
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pAddCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [836](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [887](#)
- pAddGSMSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [836](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pAddHDRSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [836](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pAddLTESysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [836](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pAddWCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)

- pAdditionalReadResult
  - unpack\_uim\_SLQSUIMReadRecord\_t, 996
- pAddrAllocPref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 127
  - LibPackprofile\_3GPP, 134
- pAddress
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pAirTimer
  - pack\_voice\_SLQSVoiceGetConfig\_t, 545
- pAirTimerCnt
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pAirTimerConfig
  - pack\_voice\_SLQSVoiceSetConfig\_t, 551
- pAirTimerStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, 1036
- pAlertType
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
- pAlertingPattern
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
- pAllowLinger
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 139
- pAlphaID
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-\_ind\_t, 949
- pAlphaIDInfo
  - unpack\_voice\_SLQSOriinateUSSD\_t, 1009
  - unpack\_voice\_SLQSVoiceDialCall\_t, 1011
  - unpack\_voice\_SLQSVoiceGetCallBarring\_t, 1017
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, 1018
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
  - unpack\_voice\_SLQSVoiceGetCallWaiting\_t, 1023
  - unpack\_voice\_SLQSVoiceGetCLIP\_t, 1024
  - unpack\_voice\_SLQSVoiceGetCLIR\_t, 1026
  - unpack\_voice\_SLQSVoiceGetCNAP\_t, 1027
  - unpack\_voice\_SLQSVoiceGetCOLP\_t, 1029
  - unpack\_voice\_SLQSVoiceGetCOLR\_t, 1030
  - unpack\_voice\_SLQSVoiceSetCallBarringPassword-\_t, 1035
  - unpack\_voice\_SLQSVoiceSetSUPSService\_t, 1038
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1041
- pAltitudeAssumed
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pAltitudeWrtEllipsoid
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pAltitudeWrtMeanSeaLevel
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pAltitudeWrtSealevel
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pAmrMode
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
- unpack\_ims\_SLQSSetIMSVoIPConfig\_t, 741
- pAmrOctetAligned
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSSetIMSVoIPConfig\_t, 741
- pAmrWBMode
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSSetIMSVoIPConfig\_t, 741
- pAmrWBOctetAligned
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSSetIMSVoIPConfig\_t, 742
- pAmrWbEnable
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSSetIMSVoIPConfig\_t, 741
- pApnString
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 139
- pApnStringSize
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 139
- pApnname
  - pack\_wds\_SetDefaultProfile\_t, 562
- pAppName
  - loc\_LocApplicationInfo, 162
- pAppPriority
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pAppProvider
  - loc\_LocApplicationInfo, 162
- pAppSubType
  - unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, 880
- pAppType
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pAppUserData
  - LibpackProfile3GPPV2, 127
- pAppVersion
  - loc\_LocApplicationInfo, 162
- pApplication
  - pack\_nas\_SetCDMANetworkParameters\_t, 429
- pApplicationInfo
  - pack\_loc\_Start\_t, 423
- pArrAlertingPattern
  - unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- pArrAlertingType
  - unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- pArrAlphaID
  - unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- pArrCallEndReason
  - unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- pArrCallInfo
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- pArrCalledPartyNum
  - unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015

- pArrConnectPartyNum
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrDiagInfo
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrRedirPartyNum
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrRemotePartyName
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrRemotePartyNum
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrSvcOption
  - unpack\_voice\_allCallStatusCallback\_ind\_t, [1003](#)
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pArrUUSInfo
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1015](#)
- pAuth
  - pack\_wds\_SLQSSStartDataSession\_t, [577](#)
- pAuthPassword
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- pAuthPassword\_tSize
  - LibPackprofile\_3GPP2, [140](#)
- pAuthPasswordSize
  - LibpackProfile3GPP2, [118](#)
- pAuthProtocol
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- pAuthRetryCount
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- pAuthStatus
  - unpack\_nas\_GetANAAAAAuthenticationStatus\_t, [794](#)
- pAuthTimeout
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- pAuthenticateResult
  - unpack\_uim\_SLQSUIMAuthenticate\_t, [990](#)
- pAuthenticationPref
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pAutoAnsStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, [1036](#)
- pAutoAnswer
  - pack\_voice\_SLQSVoiceGetConfig\_t, [546](#)
  - pack\_voice\_SLQSVoiceSetConfig\_t, [551](#)
- pAutoAnswerStat
  - unpack\_voice\_SLQSVoiceGetConfig\_t, [1032](#)
- pAutoReboot
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, [508](#)
  - unpack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, [960](#)
- pAutoSelection
  - unpack\_uim\_SLQSUIMGetConfiguration\_t, [992](#)
- pAutosdm
  - pack\_swiaoma\_SLQSOMADMSetSettings\_t, [517](#)
- pBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [847](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [849](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pBdsSVInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, [409](#)
- pBinaryUpdateSessionInfo
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, [955](#)
  - unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t, [961](#)
- pBroadcast
  - pack\_nas\_SetCDMANetworkParameters\_t, [429](#)
- pBroadcastConfig
  - pack\_sms\_SLQSSetSmsBroadcastConfig\_t, [496](#)
  - unpack\_sms\_SLQSGetSmsBroadcastConfig\_t, [939](#)
- pBurstDTMFLengths
  - pack\_voice\_SLQSVoiceBurstDTMF\_t, [540](#)
- pCCResType
  - unpack\_voice\_SLQSVoiceGetCallBarring\_t, [1017](#)
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, [1018](#)
  - unpack\_voice\_SLQSVoiceGetCallWaiting\_t, [1023](#)
  - unpack\_voice\_SLQSVoiceGetCLIP\_t, [1024](#)
  - unpack\_voice\_SLQSVoiceGetCLIR\_t, [1026](#)
  - unpack\_voice\_SLQSVoiceGetCNAP\_t, [1027](#)
  - unpack\_voice\_SLQSVoiceGetCOLP\_t, [1029](#)
  - unpack\_voice\_SLQSVoiceGetCOLR\_t, [1030](#)
  - unpack\_voice\_SLQSVoiceSetCallBarringPassword-t, [1035](#)
- pCCResultType
  - unpack\_voice\_SLQSVoiceDialCall\_t, [1011](#)
  - unpack\_voice\_SLQSVoiceSetSUPSService\_t, [1038](#)
- pCCSUPSType
  - unpack\_voice\_SLQSVoiceDialCall\_t, [1011](#)
  - unpack\_voice\_SLQSVoiceGetCallBarring\_t, [1017](#)
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, [1018](#)
  - unpack\_voice\_SLQSVoiceGetCallWaiting\_t, [1023](#)
  - unpack\_voice\_SLQSVoiceGetCLIP\_t, [1024](#)
  - unpack\_voice\_SLQSVoiceGetCLIR\_t, [1026](#)
  - unpack\_voice\_SLQSVoiceGetCNAP\_t, [1027](#)
  - unpack\_voice\_SLQSVoiceGetCOLP\_t, [1029](#)
  - unpack\_voice\_SLQSVoiceGetCOLR\_t, [1030](#)
  - unpack\_voice\_SLQSVoiceSetCallBarringPassword-t, [1035](#)
  - unpack\_voice\_SLQSVoiceSetSUPSService\_t, [1038](#)
- pCCSuppsType

- unpack\_voice\_SLQSOriinateUSSD\_t, [1009](#)
- pCDMABroadcastConfig
  - pack\_sms\_SLQSSetSmsBroadcastConfig\_t, [496](#)
  - unpack\_sms\_SLQSGetSmsBroadcastConfig\_t, [939](#)
- pCDMAChannel
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, [856](#)
- pCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pCDMAECIOTresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pCDMAECIOTreshList
  - nas\_CDMAECIOTresh, [179](#)
- pCDMAFrameErrRate
  - unpack\_nas\_SLQSGetErrorRate\_t, [814](#)
- pCDMAInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pCDMARSSITresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pCDMARSSITreshList
  - nas\_CDMARSSITresh, [181](#)
- pCDMASigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [872](#)
- pCDMASrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- PCICellInfoLen
  - nas\_QmisNasSlqsNasPCInfo, [295](#)
- pCLIPResp
  - unpack\_voice\_SLQSVoiceGetCLIP\_t, [1024](#)
- pCLIPstatus
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCLIRCause
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
- pCLIRResp
  - unpack\_voice\_SLQSVoiceGetCLIR\_t, [1026](#)
- pCLIRType
  - pack\_voice\_SLQSVoiceDialCall\_t, [541](#)
- pCLIRstatus
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCNAPResp
  - unpack\_voice\_SLQSVoiceGetCNAP\_t, [1027](#)
- pCNAPstatus
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCOLPResp
  - unpack\_voice\_SLQSVoiceGetCOLP\_t, [1029](#)
- pCOLPstatus
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCOLRResp
  - unpack\_voice\_SLQSVoiceGetCOLR\_t, [1030](#)
- pCOLRstatus
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- PCSCFAddrPCO
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)
- PCSCFFQDNAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)
- pCSCFPortName
  - pack\_ims\_SLQSSetRegMgrConfig\_t, [406](#)
- pCSCFPortNameLen
  - pack\_ims\_SLQSSetRegMgrConfig\_t, [406](#)
- PCSFlag
  - LibPackProfileMnc, [141](#)
- pCSGID
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
- pCSPPLMNModelInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- PCTlv
  - unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t, [745](#)
- pCUGIndex
  - unpack\_voice\_SUPSNotificationCallback\_ind\_t, [1044](#)
- pCUGInfo
  - pack\_voice\_SLQSVoiceDialCall\_t, [541](#)
- PCURTIv
  - unpack\_ims\_SLQSSMSCfgCallBack\_ind\_t, [750](#)
- pCallActiveDuration
  - unpack\_wds\_GetSessionDurationV2\_t, [1068](#)
- pCallBarPasswd
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1041](#)
- pCallBarringPasswd
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
- pCallFWNum
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCallFWTimerVal
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pCallForwardingNumber
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
- pCallFwdInfo
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1041](#)
- pCallFwdTypeAndPlan
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
- pCallIID
  - pack\_voice\_SLQSVoiceManageCalls\_t, [548](#)
  - pack\_voice\_SLQSVoiceSendFlash\_t, [549](#)
  - pack\_voice\_SLQSVoiceStartContDTMF\_t, [554](#)
  - unpack\_voice\_SLQSVoiceBurstDTMF\_t, [1010](#)

- unpack\_voice\_SLQSVoiceDialCall\_t, 1011
- unpack\_voice\_SLQSVoiceGetCallBarring\_t, 1017
- unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, 1018
- unpack\_voice\_SLQSVoiceGetCallWaiting\_t, 1023
- unpack\_voice\_SLQSVoiceGetCLIP\_t, 1024
- unpack\_voice\_SLQSVoiceGetCLIR\_t, 1026
- unpack\_voice\_SLQSVoiceGetCNAP\_t, 1027
- unpack\_voice\_SLQSVoiceGetCOLP\_t, 1029
- unpack\_voice\_SLQSVoiceGetCOLR\_t, 1030
- unpack\_voice\_SLQSVoiceSendFlash\_t, 1033
- unpack\_voice\_SLQSVoiceSetCallBarringPassword\_t, 1035
- unpack\_voice\_SLQSVoiceSetSUPSService\_t, 1038
- unpack\_voice\_SLQSVoiceStartContDTMF\_t, 1038
- unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1042
- voice\_burstDTMFInfo, 1121
- pCallId
  - pack\_voice\_SLQSVoiceAnswerCall\_t, 539
  - pack\_voice\_SLQSVoiceEndCall\_t, 542
  - unpack\_voice\_SLQSOOriginateUSSD\_t, 1009
  - unpack\_voice\_SLQSVoiceAnswerCall\_t, 1010
  - unpack\_voice\_SLQSVoiceEndCall\_t, 1012
- pCallInfo
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
- pCallModeStatusInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pCallPartySubAdd
  - pack\_voice\_SLQSVoiceDialCall\_t, 541
- pCallType
  - pack\_voice\_SLQSVoiceDialCall\_t, 541
- pCallWaitInd
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pCalledPartyInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pCallerIDInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pCallerNameInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pCallingPartyInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pCampedCiotLteOpMode
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 841
- pCardResult
  - unpack\_uim\_ReadTransparent\_t, 987
  - unpack\_uim\_SLQSUIMAuthenticate\_t, 990
  - unpack\_uim\_SLQSUIMGetFileAttributes\_t, 993
  - unpack\_uim\_SLQSUIMReadRecord\_t, 996
  - unpack\_uim\_SLQSUIMWriteRecord\_t, 998
  - unpack\_uim\_SLQSUIMWriteTransparent\_t, 999
  - unpack\_uim\_UnblockPinV2\_t, 1001
- pCardStatus
  - unpack\_uim\_GetCardStatus\_t, 985
  - unpack\_uim\_GetCardStatusV2\_t, 986
- unpack\_uim\_SLQSUIMSetStatusChangeCall-Back\_ind\_t, 998
- pCcResultType
  - unpack\_voice\_SLQSOOriginateUSSD\_t, 1009
- pCellDataMask
  - pack\_pds\_ResetPDSData\_t, 474
- pCellDb
  - pack\_loc\_Delete\_Assist\_Data\_t, 409
- pChangeDuration
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, 434
- pChgDuration
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
- pCiotAcqOrder
  - nas\_ciotAcqOrderPref, 187
  - nas\_CiotAcqOrderPrefTlv, 188
- pCiotAcqOrderPref
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 849
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pCiotLteOpMode
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
- pCiotLteOpModePref
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 849
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pCiotOpModePref
  - pack\_nas\_PerformNetworkScanPCI\_t, 427
- pClatFlag
  - LibpackProfile3GPPV2, 127
- pClientPerformOperationFlag
  - pack\_swiaavms\_SLQSAVMSSendSelection\_t, 504
- pClkInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, 409
- pCodecSTGain
  - pack\_audio\_SLQSSetAudioPathConfig\_t, 376
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, 657
- pColorCode
  - unpack\_nas\_SLQSNasGetHDRColorCode\_t, 863
- pConfig
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
  - unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t, 961
- pConfigAltitudeAssumed
  - pack\_loc\_Start\_t, 423
- pConfigurationMask
  - pack\_uim\_SLQSUIMGetConfiguration\_t, 525
- pConnectNumInfo
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pConnectionRequest
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
- pConnectionRetryTimers
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, 506



- pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 508
- unpack\_swiaavms\_SLQSAVMSGetSettings\_t, 958
- unpack\_swiaavms\_SLQSAVMSGetSettings\_v2\_t, 960
- pCreateProfileOut
  - unpack\_wds\_SLQSCreateProfile\_t, 1070
- pCsgld
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, 869
- pCurAMRConfig
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pCurPrefVoiceSO
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pCurProfile
  - pack\_wds\_SLQSCreateProfile\_t, 567
- pCurVoiceDomainPref
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pCurVoicePrivacyPref
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pCurrChannelRateInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pCurrDataBearerTechInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pCurrImglInfo
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 721
- pCurrPrefDataSysInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pCurrTTYMode
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pCurrentPLMNNameInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pCurrentPersonality
  - nas\_HDRPersonality\_Ind\_Data, 216
  - unpack\_nas\_SLQSSwiGetHDRPersonality\_t, 879
- pCurrentPrsnlty
  - unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, 880
- pCustSettingInfo
  - DMSgetCustomFeatureV2, 76
- pCustSettingList
  - DMSgetCustomFeatureV2, 76
- pCustomSCP
  - pack\_nas\_SetCDMANetworkParameters\_t, 429
- pCwtMute
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, 501
  - pack\_swiaudio\_SLQSSetM2MAVMute\_t, 503
- pCycleLen
  - pack\_nas\_SLQSNASSeteDRXParams\_t, 452
  - unpack\_nas\_SLQSNASGeteDRXParams\_t, 860
  - unpack\_nas\_SLQSNASGeteDRXParamsExt\_t, 862
- pDDTMInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- PDOP
  - loc\_precisionDilution, 163
- PDPTtype
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- pDRCPParams
  - unpack\_nas\_SLQSSwiGetHRPDStats\_t, 881
- pDTMFTXGain
  - pack\_audio\_SLQSSetAudioPathConfig\_t, 376
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, 657
- pData
  - pack\_cat\_CATSendEnvelopeCommand\_t, 379
  - pack\_cat\_CATSendTerminalResponse\_t, 380
- pDataBearer
  - unpack\_wds\_GetDataBearerTechnology\_t, 1052
- pDataBearerTechInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pDataCallStatusChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pDataMode
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pDataRate
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pDataSessionStatus
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
- pDataSrc
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1042
- pDataStatusDetail
  - unpack\_nas\_SLQSSwiNetworkDebug\_t, 884
- pDataSubscriptionPriorityInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pDataSystemStatusChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pDayltSavAdj
  - unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t, 870
- pDefaultPDNEnabled
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, 573
- pDeferTime
  - pack\_swiaavms\_SLQSAVMSSendSelection\_t, 504
  - pack\_swiaoma\_SLQSOMADMSendSelection\_t, 515
- pDestSMSContent
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 391
  - packgetDyingGaspCfg, 582
- pDestSMSNum
  - pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 391
  - packgetDyingGaspCfg, 582
- pDetachAction
  - pack\_nas\_SLQSSwiPSDetach\_t, 473
- pDevCurSubsCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
- pDevExplicitCfgIndex
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
- pDevMaxActDataSubsCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
- pDevMaxCfgListCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
- pDevMaxSubsCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673

- pDevMultiSimCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevMultiSimVoiceDataCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevSrvCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevSubsFeatureModeCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevSubsVoiceDataCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevVoiceCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDevVoiceDataCaps
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
- pDeviceConfigDetail
  - unpack\_nas\_SLQSSwiNetworkDebug\_t, [884](#)
- pDiagInfo
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, [1021](#)
- pDigitBuff
  - voice\_burstDTMFInfo, [1121](#)
- pDirNum
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, [856](#)
- pDisplInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
- pDnsWithDHCPFlag
  - LibpackProfile3GPPV2, [127](#)
- pDormancyStatusInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [580](#)
- pDualStandByPrefInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pDuration
  - unpack\_pds\_GetXTRAValidity\_t, [897](#)
- pDurationDueToOOS
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pDurationThreshold
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pE911StateReadyStatusInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pECIOThresList
  - nas\_ECIOThresh, [199](#)
- pECIOThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
- pECMode
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pECTNum
  - unpack\_voice\_SUPSNotificationCallback\_ind\_t, [1044](#)
- pEMBMSStatusInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- PER
  - nas\_NetworkStatEVDO, [263](#)
- pEVDOPageMonPerChangeInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, [580](#)
- pEarMute
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, [501](#)
- pEarlyWakeupTime
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pEdrxChangeInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pEdrxCiotLteMode
  - pack\_nas\_SLQSNASSeteDRXParams\_t, [452](#)
- pEdrxEnable
  - pack\_nas\_SLQSNASSeteDRXParams\_t, [453](#)
  - unpack\_nas\_SLQSNASGeteDRXParams\_t, [860](#)
  - unpack\_nas\_SLQSNASGeteDRXParamsExt\_t, [862](#)
- pEdrxRAT
  - pack\_nas\_SLQSNASGeteDRXParamsExt\_t, [440](#)
  - unpack\_nas\_SLQSNASGeteDRXParamsExt\_t, [862](#)
- pEdrxRatType
  - pack\_nas\_SLQSNASSeteDRXParams\_t, [453](#)
- pEhrpdMTUSize
  - unpack\_swidms\_SLQSSwiDmsGetMTU\_t, [965](#)
- pEmerMode
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [847](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pEmergencyCategory
  - pack\_voice\_SLQSVoiceDialCall\_t, [541](#)
- pEmergencyModeStatusInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pEnableNotification
  - pack\_wds\_DHCPv4ClientLeaseChange\_t, [555](#)
- pEnabled
  - pack\_wds\_SetMobileIPPProfile\_t, [566](#)
- pEnabledStatus
  - unpack\_pds\_GetPDSSState\_t, [894](#)
- pEncryptData
  - pack\_uim\_ReadTransparent\_t, [522](#)
- pEncryptedData
  - unpack\_uim\_ReadTransparent\_t, [987](#)
- pEncryptedPIN1
  - pack\_uim\_VerifyPin\_t, [536](#)
  - unpack\_uim\_ChangePin\_t, [985](#)
  - unpack\_uim\_SetPinProtection\_t, [988](#)
  - unpack\_uim\_UnblockPin\_t, [1000](#)
  - unpack\_uim\_UnblockPinV2\_t, [1001](#)
  - unpack\_uim\_VerifyPin\_t, [1001](#)
- pErrorRateInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pExtDisplInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
- pExtDispRecInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)

- pExtErrCode
  - UnPackGetProfileSettingOut, [1099](#)
  - UnPackGetProfileSettingOutV2, [1100](#)
- pExtErrorCode
  - unpack\_wds\_SLQSMModifyProfile\_t, [1085](#)
- pFDNStatus
  - pack\_uim\_SLQSUIIMSetServiceStatus\_t, [531](#)
  - unpack\_uim\_SLQSUIIMGetServiceStatus\_t, [993](#)
- pFailCause
  - unpack\_voice\_SLQSVoiceGetCallBarring\_t, [1017](#)
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, [1018](#)
  - unpack\_voice\_SLQSVoiceGetCallWaiting\_t, [1023](#)
  - unpack\_voice\_SLQSVoiceGetCLIP\_t, [1025](#)
  - unpack\_voice\_SLQSVoiceGetCLIR\_t, [1026](#)
  - unpack\_voice\_SLQSVoiceGetCNAP\_t, [1027](#)
  - unpack\_voice\_SLQSVoiceGetCOLP\_t, [1029](#)
  - unpack\_voice\_SLQSVoiceGetCOLR\_t, [1030](#)
  - unpack\_voice\_SLQSVoiceManageCalls\_t, [1033](#)
  - unpack\_voice\_SLQSVoiceSetCallBarringPassword-\_t, [1035](#)
  - unpack\_voice\_SLQSVoiceSetSUPSService\_t, [1038](#)
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pFailureReason
  - unpack\_wds\_SLQSSStartDataSession\_t, [1095](#)
- pFileAttributes
  - unpack\_uim\_SLQSUIIMGetFileAttributes\_t, [993](#)
- pFixId
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pFlag
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pFlashPayLd
  - pack\_voice\_SLQSVoiceSendFlash\_t, [549](#)
- pFlashType
  - pack\_voice\_SLQSVoiceSendFlash\_t, [549](#)
- pFollowOnDC
  - sms\_sendAsynsmsParams, [605](#)
- pForbiddenNetworks3GPP
  - unpack\_nas\_SLQSNASGetForbiddenNetworks\_t, [862](#)
- pForceOnDC
  - sms\_sendAsynsmsParams, [605](#)
- pForceRev0
  - pack\_nas\_SetCDMANetworkParameters\_t, [430](#)
- pFwAutoCheck
  - pack\_swiaoma\_SLQSOMADMSetSettings\_t, [517](#)
- pFwAutoSDM
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, [506](#)
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, [508](#)
- pGERANInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pGPRSMinimumQoS
  - LibpackProfile3GPP, [112](#)
- LibpackProfile3GPPV2, [127](#)
- LibPackprofile\_3GPP, [134](#)
- pGPRSRequestedQoS
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [127](#)
  - LibPackprofile\_3GPP, [134](#)
- pGPSDataMask
  - pack\_pds\_ResetPDSDData\_t, [475](#)
- pGPSWeek
  - unpack\_pds\_GetXTRAVValidity\_t, [897](#)
- pGPSWeekOffset
  - unpack\_pds\_GetXTRAVValidity\_t, [897](#)
- pGSMBER
  - unpack\_nas\_SLQSGetErrorRate\_t, [814](#)
- pGSMCallBarringSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pGSMCipherDomainSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pGSMRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pGSMRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [437](#)
- pGSMRSSIThreshList
  - nas\_GSMRSSIThresh, [210](#)
- pGSMSigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [872](#)
- pGSMSSrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pGSMSSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [841](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pGWAcqOrderPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [847](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pGcellInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pGenerator
  - pack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [498](#)
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, [501](#)
- pGetCallFWExtInfo
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, [1019](#)
- pGetCallFWInfo
  - unpack\_voice\_SLQSVoiceGetCallForwarding-Status\_t, [1019](#)
- pGetCustomInput
  - DMSGetCustomFeatureV2, [76](#)



- pGetDyingGaspCfg
  - unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, 720
- pGetDyingGaspStatistics
  - unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, 720
- pGetIndicationRegInfo
  - unpack\_sms\_SLQSGetIndicationRegister\_t, 937
- pGetMsgWaitingInfoResp
  - unpack\_sms\_SLQSGetMessageWaiting\_t, 937
- pGetSlotsInfoTlv
  - unpack\_uim\_SLQSUIMGetSlotsStatusV2\_t, 995
- pGetSlotsStatusTlv
  - unpack\_uim\_SLQSUIMGetSlotsStatusV2\_t, 995
- pGetTransLayerInfo
  - unpack\_sms\_SLQSGetTransLayerInfo\_t, 940
- pGetTransNWRegInfo
  - unpack\_sms\_SLQSGetTransNWRegInfo\_t, 941
- pGnssData
  - pack\_loc\_Delete\_Assist\_Data\_t, 409
- pGpsTime
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pGyroAcceptReady
  - unpack\_loc\_SensorStreamingCallback\_Ind\_t, 788
- pGyroSamplesAccepted
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, 779
- pGyroTempAcceptReady
  - unpack\_loc\_SensorStreamingCallback\_Ind\_t, 788
- pGyroTempSamplesAccepted
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, 779
- pHA2002bis
  - pack\_wds\_SetMobileIPParameters\_t, 564
- pHAAAuthenticator
  - pack\_wds\_SetMobileIPParameters\_t, 564
- pHASPI
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pHDRECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 437
- pHDRECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 437
- pHDRECIOThreshList
  - nas\_HDRECIOThresh, 214
- pHDRIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHDRIOTThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHDRIOTThreshList
  - nas\_HDRIOTThresh, 215
- pHDRNewUATIAssInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pHDRPackErrRate
  - unpack\_nas\_SLQSGetErrorRate\_t, 814
- pHDRPersInd
  - unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind\_t, 883
- pHRRSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHRRSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHRRSSIThreshList
  - nas\_HHRRSSIThresh, 216
- pHRSINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHRSINRThresList
  - nas\_HHRSINRThresh, 217
- pHRSINRThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, 432
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pHRSINRThreshList
  - nas\_HHRSINRThreshold, 217
- pHRSsessionCloseInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pHRSigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 873
- pHRSrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 841
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pHRSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 841
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pHTTPStatus
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
- pHWWWatchdog
  - unpack\_swidms\_SLQSSwiDmsGetHWWWatchdog\_t, 964
- pHaltSubscription
  - unpack\_uim\_SLQSUIMGetConfiguration\_t, 992
- pHeading
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHeadingUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHiddenKeyStatus
  - unpack\_uim\_SLQSUIMGetServiceStatus\_t, 993
- pHomeNwMNC3Gpp
  - unpack\_nas\_SLQSGetHomeNetwork\_t, 816
- pHomeSIDNID
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, 856
- pHorCirConf
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
- pHorConfidence
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorEllpConf
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
- pHorReliability
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorUncCircular
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766

- unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorUncEllipseOrientAzimuth
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorUncEllipseSemiMajor
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorUncEllipseSemiMinor
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pHorizontalAccuracyLvl
  - pack\_loc\_Start\_t, 423
- pHorizontalConfidence
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pHorizontalUncCircular
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pHotSwapStatus
  - unpack\_uim\_GetCardStatus\_t, 985
  - unpack\_uim\_GetCardStatusV2\_t, 986
- pHrpdMTUSize
  - unpack\_swidms\_SLQSSwiDmsGetMTU\_t, 965
- pHwConfig
  - pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t, 576
  - unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, 1094
- PI
  - voice\_calledPartyInfo, 1122
  - voice\_callerIDInfo, 1124
  - voice\_callFWExtInfo, 1127
  - voice\_callingPartyInfo, 1131
  - voice\_redirNumInfo, 1151
- PIFACEId
  - pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, 500
- pIMSDomain
  - pack\_ims\_SLQSSetIMSUserConfig\_t, 403
  - unpack\_ims\_SLQSSetIMSUserConfig\_t, 739
- pIMSDomainLen
  - pack\_ims\_SLQSSetIMSUserConfig\_t, 403
  - unpack\_ims\_SLQSSetIMSUserConfig\_t, 739
- pIMSPrefStatusInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 450
- pIMSTestMode
  - pack\_ims\_SLQSSetRegMgrConfig\_t, 406
  - unpack\_ims\_SLQSSetRegMgrConfig\_t, 743
- pIOThresList
  - nas\_IOThresh, 223
- pIOThresh
  - pack\_nas\_SLQSSetConfigSigInfo\_t, 432
- pIPv6DelegFlag
  - LibpackProfile3GPPV2, 128
- pIPv4AddrInfo
  - unpack\_loc\_GetServer\_Ind\_t, 775
- pIPv4AddrPref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pIPv4Config
  - pack\_loc\_SLQSLOCSetServer\_t, 422
- pIPv6AddPref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pIPv6AddrInfo
  - unpack\_loc\_GetServer\_Ind\_t, 775
- pIPv6Config
  - pack\_loc\_SLQSLOCSetServer\_t, 422
- pIWLANtoLTEHandoverFlag
  - LibpackProfile3GPPV2, 128
- plgnoreHotSwapSwitch
  - pack\_uim\_SLQSUIPowerUp\_t, 528
- plmCnFlag
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- plmImageList
  - pack\_fms\_SetImagesPreference\_t, 400
  - unpack\_fms\_GetImagesPreference\_t, 736
- plmsVoiceSupportLte
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 841
- plIndex
  - unpack\_uim\_SLQSUIGetServiceStatus\_t, 993
- plIndicationToken
  - pack\_uim\_ChangePin\_t, 521
  - pack\_uim\_ReadTransparent\_t, 522
  - pack\_uim\_SetPinProtection\_t, 523
  - pack\_uim\_SLQSUIAuthenticate\_t, 524
  - pack\_uim\_SLQSUIGetFileAttributes\_t, 526
  - pack\_uim\_SLQSUIReadRecord\_t, 529
  - pack\_uim\_SLQSUIWriteRecord\_t, 533
  - pack\_uim\_SLQSUIWriteTransparent\_t, 534
  - pack\_uim\_UnblockPin\_t, 535
  - pack\_uim\_VerifyPin\_t, 536
  - unpack\_uim\_ChangePin\_t, 985
  - unpack\_uim\_ReadTransparent\_t, 987
  - unpack\_uim\_SetPinProtection\_t, 988
  - unpack\_uim\_SLQSUIAuthenticate\_t, 990
  - unpack\_uim\_SLQSUIGetFileAttributes\_t, 993
  - unpack\_uim\_SLQSUIReadRecord\_t, 996
  - unpack\_uim\_SLQSUIWriteRecord\_t, 998
  - unpack\_uim\_SLQSUIWriteTransparent\_t, 999
  - unpack\_uim\_UnblockPin\_t, 1000
  - unpack\_uim\_UnblockPinV2\_t, 1001
  - unpack\_uim\_VerifyPin\_t, 1001
- plInfo
  - pack\_voice\_AnswerUSSD\_t, 536
  - pack\_voice\_OriginateUSSD\_t, 537
- plInstanceSize
  - nas\_RfBandInfoList, 301
- plInstancesSize
  - nas\_RfBandInfoExtFormat, 299
  - nas\_RfBandwidthInfo, 303
  - nas\_RfDedicatedBandInfo, 306
- plInterfaceCfg
  - unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t, 967
- plIntermediateReportState

- pack\_loc\_Start\_t, [423](#)
- pInterval
  - unpack\_pds\_GetPDSDefaults\_t, [893](#)
  - unpack\_pds\_GetXTRAAutomaticDownload\_t, [896](#)
- plpcpAckTimeout
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- plpcpCreqRetryCount
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- plsPscfAddressNedded
  - LibpackProfile3GPP2, [118](#)
  - LibPackprofile\_3GPP2, [140](#)
- pKeyReferenceID
  - pack\_uim\_ChangePin\_t, [521](#)
  - pack\_uim\_SetPinProtection\_t, [523](#)
  - pack\_uim\_UnblockPin\_t, [535](#)
  - pack\_uim\_VerifyPin\_t, [536](#)
- PLMNData
  - nas\_operatorPLMNList, [279](#)
- PLMNName
  - nas\_operatorNameString, [277](#)
- PLMNNetName
  - nas\_PLMNNetworkName, [286](#)
- PLMNRecID
  - nas\_OperatorPLMNData, [278](#)
- pLTEAttachProfile
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [573](#)
- pLTEAttachProfileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [573](#)
- pLTEBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [847](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pLTEBandPrefExt
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [472](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pLTCphyCa
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [450](#)
- pLTEInfo
  - unpack\_nas\_SLQSNasSwiModemStatus\_t, [874](#)
- pLTEInfoInterfreq
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pLTEInfoIntrafreq
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pLTEInfoNeighboringGSM
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
- unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pLTEInfoNeighboringWCDMA
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pLTEOperationMode
  - nas\_LTEOperationMode, [241](#)
  - unpack\_nas\_SLQSNasGetRFInfo\_t, [864](#)
- pLTERACHFailInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pLTERRCTxInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pLTERSRPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSRPThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSRPThreshList
  - nas\_LTERSRPThresh, [245](#)
- pLTERSRQDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSRQThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSRQThreshList
  - nas\_LTERSRQThresh, [245](#)
- pLTERSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTERSSIThreshList
  - nas\_LTERSSIThresh, [246](#)
- pLTESIB16NetworkTimeInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pLTESNRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTESNRThresList
  - nas\_LTESNRThresh, [249](#)
- pLTESNRThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTESNRThreshList
  - nas\_LTESNRThreshold, [249](#)
- pLTESigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [873](#)
- pLTESigRptCfg
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
- pLTESigRptConfig
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pLTESrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [842](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pLTESysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [842](#)
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pLTEVoiceSupportSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, [837](#)

- unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pLTetoWLANHandoverFlag
  - LibpackProfile3GPPV2, 128
- pLastCallActiveDuration
  - unpack\_wds\_GetSessionDurationV2\_t, 1068
- pLastCallDuration
  - unpack\_wds\_GetSessionDurationV2\_t, 1068
- pLastRecord
  - pack\_uim\_SLQSUIReadRecord\_t, 529
- pLatitude
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pLcpAckTimeout
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pLcpCreqRetryCount
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pLeapSeconds
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pLineCtrlInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pLinktimer
  - pack\_sms\_SendSMS\_t, 489
  - sms\_sendAsynsmsParams, 605
- pLongitude
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pLteCiotOpModeTlv
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pLteEARFCN
  - pack\_nas\_SLQSNASSwiSetChannelLock\_t, 454
  - unpack\_nas\_SLQSNASSwiGetChannelLock\_t, 873
- pLteEarfcnInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, 859
- pLteEmbmsCoverage
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 841
- pLteEmbmsTraceId
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pLteM1BandPref
  - pack\_nas\_PerformNetworkScanPCI\_t, 427
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pLteNB1BandPref
  - pack\_nas\_PerformNetworkScanPCI\_t, 428
- pLteNb1BandPref
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pLteOpMode
  - pack\_nas\_SLQSNASGeteDRXParamsExt\_t, 440
  - unpack\_nas\_SLQSNASGeteDRXParamsExt\_t, 862
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, 869
- pLteOpModeTlv
  - unpack\_nas\_PerformNetworkScan\_t, 808
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pLtePCI
  - pack\_nas\_SLQSNASSwiSetChannelLock\_t, 454
  - unpack\_nas\_SLQSNASSwiGetChannelLock\_t, 873
- pLteRegDomain
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pLteRoamPDPTYPE
  - LibpackProfile3GPPV2, 128
- pLteVoiceDomain
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pMCC
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pMICGainSelect
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, 657
- pMIPStatusInd
  - pack\_wds\_SLQSWdsSetEventReport\_t, 580
- pMNA
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pMNC
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pMNCIncPCSDigStat
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
- pMNHA
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pMNRInfo
  - pack\_nas\_SLQSIInitiateNetworkRegistration\_t, 434
- pMTUSize3gpp
  - unpack\_swidms\_SLQSSwiDmsGetMTU\_t, 965
- pMagneticDeviation
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pManagedRoamingInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pManualNWScanFailureInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pMaxPDN
  - LibpackProfile3GPPV2, 128
- pMaxPDNTimer
  - LibpackProfile3GPPV2, 128
- pMaxStorageSizeReq
  - pack\_sms\_SLQSSmsGetMaxStorageSize\_t, 497
- pMaxStorageSizeResp
  - unpack\_sms\_SLQSSmsGetMaxStorageSize\_t, 945
- pMcc
  - LibpackProfile3GPPV2, 128
- pMdn

- unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, 856
- pMessage
  - pack\_sms\_SaveSMS\_t, 489
  - pack\_sms\_SendSMS\_t, 490
  - sms\_sendAsynsmsParams, 605
- pMessageIndex
  - pack\_sms\_SLQSDDeleteSMS\_t, 491
  - unpack\_sms\_SaveSMS\_t, 933
- pMessageMode
  - pack\_sms\_SLQSDDeleteSMS\_t, 491
  - pack\_sms\_SLQSGetSMS\_t, 492
  - pack\_sms\_SLQSGetSMSList\_t, 493
  - pack\_sms\_SLQSMModifySMSStatus\_t, 494
  - sms\_maxStorageSizeReq, 599
- pMessageProtocol
  - unpack\_sms\_SLQSSmsGetMessageProtocol\_t, 945
- pMessageTag
  - pack\_sms\_SLQSDDeleteSMS\_t, 491
- pMicMute
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, 501
- pMinBasedIMSI
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, 856
- pMinIntervalTime
  - pack\_loc\_Start\_t, 423
- pMinSessionExpiryTimer
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, 742
- pMnc
  - LibpackProfile3GPPV2, 128
- pMncPcsDigitStatus
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, 434
- pMncPcsStatus
  - pack\_nas\_SLQSGetPLMNName\_t, 433
- pMode
  - pack\_wds\_SetMobileIPParameters\_t, 564
  - unpack\_loc\_GetOpMode\_Ind\_t, 774
- pModePref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 847
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pMsisdnFlag
  - LibpackProfile3GPPV2, 128
- pMuxID
  - pack\_qos\_BindDataPort\_t, 486
- pNAI
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pNAMNameInfo
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, 856
- pNID
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNITZInformation
  - unpack\_nas\_SLQSGetOperatorNameData\_t, 818
- pNR5GCellStatus
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pNR5GCellStatusInfo
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pNR5GSerStatTlv
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pNR5GSrvStatusinfo
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pNR5GSysInfo
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
- pNR5GSystemInfoTlv
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pNSEnable
  - pack\_audio\_SLQSSetAudioPathConfig\_t, 376
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, 657
- pNSSAudioCtrl
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- pNSSRelease
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- PNTlv
  - unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t, 745
- pNamID
  - pack\_voice\_SLQSVoiceGetConfig\_t, 546
- pName
  - pack\_wds\_SetDefaultProfile\_t, 562
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNegoDnsSrvrPref
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pNeighborSetPilotPN
  - nas\_NetworkStat1x, 261
- pNetSelPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 847
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pNetworkMode
  - pack\_pds\_SLQSGetAGPSConfig\_t, 479
  - pack\_pds\_SLQSSetAGPSConfig\_t, 482
- pNetworkNameSrcTlv
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pNetworkRejectInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pNetworkStat1x
  - unpack\_nas\_SLQSSwiNetworkDebug\_t, 884
- pNetworkStatEVDO
  - unpack\_nas\_SLQSSwiNetworkDebug\_t, 884
- pNetworkTimeInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, 443
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pNetworkType
  - pack\_nas\_PerformNetworkScanPCI\_t, 428
- pNewPwdData



- unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1042
- pNotifStore
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, 506
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 508
- pNotification
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
  - unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t, 962
- pNotificationStore
  - unpack\_swiaavms\_SLQSAVMSGetSettings\_t, 958
  - unpack\_swiaavms\_SLQSAVMSGetSettings\_v2\_t, 960
- pNr5gBandPref
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pNumberOfPhySlot
  - unpack\_uim\_SLQSUIMGetSlotsStatus\_t, 994
- pNw2DescDisp
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNw2DescEnc
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNw2DescLen
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNw2MCC
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNw2MNC
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNw2Name
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pNwNameSrc3Gpp
  - unpack\_nas\_SLQSGetHomeNetwork\_t, 816
- pOTASPStatus
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
- pObjectVer
  - unpack\_nas\_SLQSSwiNetworkDebug\_t, 884
- pOffLength
  - unpack\_voice\_DTMFEventCallback\_ind\_t, 1005
- pOnLength
  - unpack\_voice\_DTMFEventCallback\_ind\_t, 1005
- pOpaqueIdentifier
  - unpack\_loc\_InjectSensorDataCallback\_Ind\_t, 779
- pOperation
  - unpack\_pds\_GetPDSDDefaults\_t, 893
- pOperatorNameDataInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pOperatorNameString
  - unpack\_nas\_SLQSGetOperatorNameData\_t, 818
- pOperatorPCOID
  - LibpackProfile3GPPV2, 128
- pOperatorPLMNList
  - unpack\_nas\_SLQSGetOperatorNameData\_t, 818
- pOptList
  - wds\_DHCPv4OptionList, 1166
- wdsDhcpv4OptionList, 1181
- pOptVal
  - wds\_DHCPOpt, 1164
- pOverridePDPTType
  - LibpackProfile3GPPV2, 128
- pPCIInfo
  - unpack\_nas\_PerformNetworkScan\_t, 808
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pPCMPParams
  - pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, 500
- pPCOIDList
  - LibpackProfile3GPPV2, 128
- pPCSCFPort
  - unpack\_ims\_SLQSGetRegMgrConfig\_t, 743
- pPCSInstance
  - unpack\_nas\_PerformNetworkScan\_t, 808
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pPCSInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, 808
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pPDNDDisconnectWaitTimer
  - LibpackProfile3GPPV2, 128
- pPDNInactivTimeout
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPDNInactivTimeout3GPP2
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pPDNThrottleTimer
  - LibpackProfile3GPPV2, 128
- pPDNWaitTimer
  - LibpackProfile3GPPV2, 128
- pPDPTType
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 135
- pPLMNNetworkName
  - unpack\_nas\_SLQSGetOperatorNameData\_t, 818
- pPRLPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 472
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 847
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pPWD
  - PackSwiAVMSSettingsAPNInfo, 585
- pPackageID
  - pack\_swiaavms\_SLQSAVMSSendSelection\_t, 504
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
  - unpack\_swiaavms\_SLQSAVMSSessionGetInfo\_t, 962
- pPagingTimeWindow
  - pack\_nas\_SLQSNASSeteDRXParams\_t, 453
  - unpack\_nas\_SLQSNASGeteDRXParams\_t, 860

- unpack\_nas\_SLQSNASGeteDRXParamsExt\_t, 862
- pPass
  - pack\_wds\_SLQSSStartDataSession\_t, 577
- pPassword
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
  - pack\_wds\_SetDefaultProfile\_t, 562
- pPasswordSize
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPcscfAddrUsingDhcp
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPcscfAddrUsingPCO
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPdnType
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pPdpAccessConFlag
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPdpContext
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPdpDataCompType
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPdpHdrCompType
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 128
  - LibPackprofile\_3GPP, 134
- pPeriodInfo
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, 506
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 508
- pPeriodicUpdateTimer
  - pack\_dms\_SLQSSetPowerSaveModeConfig\_t, 390
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, 717
- pPeripheralEndPointID
  - pack\_qos\_BindDataPort\_t, 486
- pPeroidsInfo
  - unpack\_swiaavms\_SLQSAVMSSetSettings\_t, 958
  - unpack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 960
- pPersistFlag
  - LibpackProfile3GPPV2, 129
- pPersonalityListLength
  - nas\_HDRPersonality\_Ind\_Data, 216
- unpack\_nas\_SLQSSwiGetHDRPersonality\_t, 879
- unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, 880
- pPersonalizationStatus
  - unpack\_uim\_SLQSUIMGetConfiguration\_t, 992
- pPhoneCtxtURI
  - pack\_ims\_SLQSSetIMSSMSConfig\_t, 402
  - unpack\_ims\_SLQSGetIMSSMSConfig\_t, 738
- pPhoneCtxtURILen
  - pack\_ims\_SLQSSetIMSSMSConfig\_t, 402
  - unpack\_ims\_SLQSGetIMSSMSConfig\_t, 738
- pPilotSetData
  - unpack\_nas\_SLQSSwiGetHRPDStats\_t, 881
- pPilotSetInfo
  - nas\_PilotSetData, 284
- pPlmnlId
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, 869
- pPollingTimer
  - pack\_swiaavms\_SLQSAVMSSetSettings\_t, 506
  - pack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 508
  - unpack\_swiaavms\_SLQSAVMSSetSettings\_t, 958
  - unpack\_swiaavms\_SLQSAVMSSetSettings\_v2\_t, 960
- pPositionSource
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pPppSessCloseTimer1x
  - LibpackProfile3GPP2, 118
  - LibPackprofile\_3GPP2, 140
- pPppSessCloseTimerDO
  - LibpackProfile3GPP2, 119
  - LibPackprofile\_3GPP2, 140
- pPrecisionDilution
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 786
- pPrefVoiceDomain
  - pack\_voice\_SLQSVoiceSetConfig\_t, 551
- pPrefVoicePrivacy
  - pack\_voice\_SLQSVoiceGetConfig\_t, 546
- pPrefVoiceSO
  - pack\_voice\_SLQSVoiceGetConfig\_t, 546
  - pack\_voice\_SLQSVoiceSetConfig\_t, 551
- pPrefVoiceSOStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, 1036
- pPreference
  - unpack\_pds\_GetXTRANetwork\_t, 896
- pPriCSCFPort
  - pack\_ims\_SLQSSetRegMgrConfig\_t, 406
- pPriCSCFPortName
  - unpack\_ims\_SLQSGetRegMgrConfig\_t, 743
- pPriCSCFPortNameLen
  - unpack\_ims\_SLQSGetRegMgrConfig\_t, 743
- pPriDNSIPv4AddPref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 129
  - LibPackprofile\_3GPP, 135
- pPriDNSIPv6addpref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 129

- LibPackprofile\_3GPP, [135](#)
- pPriV6DnsAddress
  - LibpackProfile3GPP2, [119](#)
  - LibPackprofile\_3GPP2, [140](#)
- pPrimaryHA
  - pack\_wds\_SetMobileIPProfile\_t, [566](#)
- pPrimaryID
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pPrimaryV4DnsAddress
  - LibpackProfile3GPP2, [119](#)
  - LibPackprofile\_3GPP2, [140](#)
- pProfileChangeInd
  - pack\_wds\_SLQSSwiProfileChangeCallback\_t, [578](#)
- pProfileID
  - unpack\_wds\_SLQSCreateProfile\_t, [1070](#)
- pProfileId
  - pack\_wds\_SLQSCreateProfile\_t, [567](#)
  - pack\_wds\_SLQSModifyProfile\_t, [572](#)
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [575](#)
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [576](#)
- pProfileList
  - pack\_wds\_SLQSSet3GPPConfigItem\_t, [573](#)
- pProfileSettings
  - unpack\_wds\_SLQSGetProfileSettings\_t, [1080](#)
  - unpack\_wds\_SLQSGetProfileSettingsV2\_t, [1081](#)
- pProfileType
  - pack\_wds\_SLQSCreateProfile\_t, [567](#)
  - pack\_wds\_SLQSModifyProfile\_t, [572](#)
- pProfilename
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pProfilenameSize
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pProtoSubTypeElmnt
  - unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, [880](#)
- pProtocol
  - pack\_nas\_SetCDMANetworkParameters\_t, [430](#)
- pProtocolSubtypeElement
  - nas\_HDRPersonality\_Ind\_Data, [216](#)
  - unpack\_nas\_SLQSSwiGetHDRPersonality\_t, [879](#)
- pPsmEnableState
  - pack\_dms\_SLQSSetPowerSaveModeConfig\_t, [390](#)
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pQosClassID
  - LibpackProfile3GPP, [112](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pRAT
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [473](#)
- pRATInstance
  - unpack\_nas\_PerformNetworkScan\_t, [808](#)
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, [877](#)
- pRATInstanceSize
  - unpack\_nas\_PerformNetworkScan\_t, [808](#)
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, [877](#)
- pRATType
  - LibpackProfile3GPP2, [119](#)
  - LibPackprofile\_3GPP2, [140](#)
- pRFBandInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pRFBandInfoParam
  - nas\_RfBandInfoList, [301](#)
- PRIStrng
  - unpack\_dms\_GetFirmwareRevision\_t, [679](#)
  - unpack\_dms\_GetFirmwareRevisions\_t, [681](#)
- PRLEnable
  - pack\_swima\_SLQSOMADMSetSettingsExt\_t, [518](#)
- PRLInd
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [831](#)
- PRLPTlv
  - NASQmiCbkNasSystemSelPrefInd, [368](#)
- PRLPref
  - nas\_PRLPrefTlv, [289](#)
  - NASPRLPreferenceTlv, [365](#)
- pRSRPTThresList
  - nas\_RSRPThresh, [311](#)
- pRSRPTThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
- pRSRQThresList
  - nas\_RSRQThresh, [312](#)
- pRSRQThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
- pRSSIThresList
  - nas\_RSSIThresh, [313](#)
- pRSSIThresh
  - pack\_nas\_SLQSConfigSigInfo\_t, [432](#)
- pRTPRTCPInactTimer
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, [405](#)
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, [742](#)
- pRTREConfigurationInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pRXAGCList
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pRXAIG
  - audio\_RXAGCList, [42](#)
- pRXAVCAGCSwitch
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pRXAVCList
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pRXChain0Info
  - unpack\_nas\_SLQSNasGetTxRxInfo\_t, [866](#)
- pRXChain1Info
  - unpack\_nas\_SLQSNasGetTxRxInfo\_t, [866](#)



- pRXComprSlope
  - audio\_RXAGCList, [43](#)
- pRXComprThres
  - audio\_RXAGCList, [43](#)
- pRXDroppedCount
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXExpSlope
  - audio\_RXAGCList, [43](#)
- pRXExpThres
  - audio\_RXAGCList, [43](#)
- pRXOKBytesLastCall
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXOkBytesCount
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXPCMIIRFtr
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pRXPacketErrors
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXPacketOverflows
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXPacketSuccesses
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pRXStaticGain
  - audio\_RXAGCList, [43](#)
- pRXTotalBytes
  - unpack\_wds\_GetByteTotals\_t, [1050](#)
- pRadioInterface
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, [870](#)
- pRandomizationWindow
  - unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, [717](#)
- pRankIndicatorInd
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- pRatDisabledMask
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pReRegPeriod
  - pack\_wds\_SetMobileIPParameters\_t, [564](#)
- pReRegTraffic
  - pack\_wds\_SetMobileIPParameters\_t, [564](#)
- pReadResult
  - unpack\_uim\_ReadTransparent\_t, [987](#)
  - unpack\_uim\_SLQSUIReadRecord\_t, [996](#)
- pReason
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pReasonMask
  - unpack\_dms\_GetOfflineReason\_t, [687](#)
- pRecurrenceType
  - pack\_loc\_Start\_t, [423](#)
- pRedirNumInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
- pRefreshEvent
  - unpack\_uim\_SLQSUIRefreshGetLastEvent\_t, [997](#)
- pRegCallStatInfoEvt
  - sms\_getIndicationReg, [597](#)
  - sms\_setIndicationReg, [606](#)
- pRegDTMFEvents
  - pack\_voice\_SLQSVoiceIndicationRegister\_t, [547](#)
- pRegInd
  - sms\_getTransLayerInfo, [598](#)
- pRegMgrConfigEvents
  - pack\_ims\_SLQSImsConfigIndicationRegister\_t, [401](#)
- pRegStatus
  - sms\_getTransNWRegInfo, [598](#)
  - unpack\_swiaims\_SLQSAVMSEventReportInd\_t, [955](#)
- pRegTransLayerInfoEvt
  - sms\_getIndicationReg, [597](#)
  - sms\_setIndicationReg, [606](#)
- pRegTransNWRegInfoEvt
  - sms\_getIndicationReg, [597](#)
  - sms\_setIndicationReg, [606](#)
- pRegVoicePrivacyEvents
  - pack\_voice\_SLQSVoiceIndicationRegister\_t, [547](#)
- pRejectReason
  - pack\_swiaims\_SLQSAVMSSendSelection\_t, [504](#)
  - pack\_swioama\_SLQSOMADMSendSelection\_t, [515](#)
- pRemainingRetries
  - unpack\_uim\_ChangePin\_t, [985](#)
  - unpack\_uim\_SetPinProtection\_t, [988](#)
  - unpack\_uim\_SLQSUIIDepersonalization\_t, [990](#)
  - unpack\_uim\_UnblockPin\_t, [1000](#)
  - unpack\_uim\_UnblockPinV2\_t, [1001](#)
  - unpack\_uim\_VerifyPin\_t, [1002](#)
- pRemotePartyName
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, [1021](#)
- pRemotePartyNum
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, [1021](#)
- pReportChannelRate
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [569](#)
- pReportConnStatus
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [569](#)
- pReportDataBearerTech
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [569](#)
- pReportDormStatus
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [569](#)
- pReqSettings
  - pack\_wds\_SLQSGetRuntimeSettings\_t, [571](#)
- pRequestOptionList
  - pack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [576](#)
  - unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, [1094](#)
- pRequestedTag
  - pack\_sms\_SLQSGetSMSList\_t, [493](#)
- pRetryInterval
  - pack\_wds\_SetMobileIPParameters\_t, [564](#)
- pRetryLimit
  - pack\_wds\_SetMobileIPParameters\_t, [564](#)
- pRetryMessage

- sms\_sendAsynsmsParams, 605
- pRetryMessageld
  - sms\_sendAsynsmsParams, 605
- pRevInUse
  - nas\_CDMASysInfo, 185
- pRevInUseValid
  - nas\_CDMASysInfo, 185
- pRevTunneling
  - pack\_wds\_SetMobileIPProfile\_t, 566
- pRfBandInfoExtFormat
  - unpack\_nas\_SLQSNasGetRFInfo\_t, 864
- pRfBandInfoExtFormatParam
  - nas\_RfBandInfoExtFormat, 299
- pRfBandwidthInfo
  - unpack\_nas\_SLQSNasGetRFInfo\_t, 864
- pRfBandwidthInfoParam
  - nas\_RfBandwidthInfo, 303
- pRfDedicatedBandInfo
  - unpack\_nas\_SLQSNasGetRFInfo\_t, 864
- pRfDedicatedBandInfoParam
  - nas\_RfDedicatedBandInfo, 306
- pRingBackTimer
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, 742
- pRingTimer
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, 742
- pRoamDisallowFlag
  - LibpackProfile3GPPV2, 129
- pRoamPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, 462
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 473
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, 847
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, 850
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- pRoamTimer
  - pack\_voice\_SLQSVoiceGetConfig\_t, 546
- pRoamTimerCnt
  - unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- pRoamTimerConfig
  - pack\_voice\_SLQSVoiceSetConfig\_t, 551
- pRoamTimerStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, 1036
- pRoaming
  - pack\_nas\_SetCDMANetworkParameters\_t, 430
- pRscp
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 873
- PSDomain
  - unpack\_nas\_GetServingNetwork\_t, 805
- pSID
  - unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- pSiODataPort
  - pack\_qos\_BindDataPort\_t, 486
- pSIPConfigEvents
  - pack\_ims\_SLQSImsConfigIndicationRegister\_t, 401
- pSIPLocalPort
  - pack\_ims\_SLQSSetSIPConfig\_t, 407
  - unpack\_ims\_SLQSGetSIPConfig\_t, 744
- PSMCFGChangeInfo
  - pack\_dms\_SetIndicationRegister\_t, 387
- pSMSAttemptedFlag
  - packgetDyingGaspStatistics, 582
- pSMSCAddress
  - pack\_sms\_SetSMSCAddress\_t, 490
  - unpack\_sms\_GetSMSCAddress\_t, 932
- pSMSCType
  - pack\_sms\_SetSMSCAddress\_t, 490
  - unpack\_sms\_GetSMSCAddress\_t, 932
- pSMSCConfigEvents
  - pack\_ims\_SLQSImsConfigIndicationRegister\_t, 401
- pSMSFormat
  - pack\_ims\_SLQSSetIMSSMSConfig\_t, 402
  - unpack\_ims\_SLQSGetIMSSMSConfig\_t, 738
- pSMSOverIPNwInd
  - pack\_ims\_SLQSSetIMSSMSConfig\_t, 402
  - unpack\_ims\_SLQSGetIMSSMSConfig\_t, 738
- PSMStatus
  - pack\_dms\_SetIndicationRegister\_t, 387
- pSPC
  - pack\_nas\_SetCDMANetworkParameters\_t, 430
  - pack\_wds\_SetMobileIPParameters\_t, 564
- pSSACBarringExtInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pSSACBarringInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, 451
- pSV
  - loc\_BdsSVInfo, 151
  - loc\_SVInfo, 168
- pSVInfo
  - pack\_loc\_Delete\_Assist\_Data\_t, 409
- pSarRFState
  - unpack\_sar\_SLQSGetRfSarState\_t, 931
- pSatelliteInfo
  - unpack\_loc\_GnssSvInfo\_Ind\_t, 776
- pScanResult
  - unpack\_nas\_PerformNetworkScan\_t, 808
  - unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- pScanType
  - pack\_nas\_PerformNetworkScanPCI\_t, 428
- pScCRxInfo
  - unpack\_nas\_SLQSSwiGetLteScCRxInfo\_t, 882
- pScrAmrEnable
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, 742
- pScrAmrWbEnable
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, 405
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, 742
- pSecDNSIPv4AddPref
  - LibpackProfile3GPP, 112
  - LibpackProfile3GPPV2, 129
  - LibPackprofile\_3GPP, 135
- pSecDNSIPv6addpref
  - LibpackProfile3GPP, 113

- LibpackProfile3GPPV2, [129](#)
- LibPackprofile\_3GPP, [135](#)
- pSecV6DnsAddress
  - LibpackProfile3GPP2, [119](#)
  - LibPackprofile\_3GPP2, [140](#)
- pSecondaryFlag
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pSecondaryHA
  - pack\_wds\_SetMobileIPProfile\_t, [566](#)
- pSecondaryV4DnsAddress
  - LibpackProfile3GPP2, [119](#)
  - LibPackprofile\_3GPP2, [140](#)
- pSectorID
  - nas\_NetworkStatEVDO, [263](#)
- pSendSmsParams
  - pack\_sms\_SLQSSendAsyncSMS\_t, [494](#)
- pSensorDataUsage
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pServerAddress
  - pack\_pds\_SLQSSetAGPSConfig\_t, [482](#)
  - unpack\_pds\_SLQSGetAGPSConfig\_t, [901](#)
- pServerPort
  - pack\_pds\_SLQSSetAGPSConfig\_t, [482](#)
  - unpack\_pds\_SLQSGetAGPSConfig\_t, [901](#)
- pServerURL
  - pack\_pds\_SLQSSetAGPSConfig\_t, [482](#)
  - unpack\_pds\_SLQSGetAGPSConfig\_t, [901](#)
- pServerURLLength
  - pack\_pds\_SLQSSetAGPSConfig\_t, [482](#)
  - unpack\_pds\_SLQSGetAGPSConfig\_t, [901](#)
- pServiceClass
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
- pServiceOption
  - sms\_sendAsynsmsParams, [605](#)
- pServingSystemInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSessionExpiryTimer
  - pack\_ims\_SLQSSetIMSVoIPConfig\_t, [405](#)
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, [742](#)
- pSessionType
  - unpack\_swiaVms\_SLQSAVMSEventReportInd\_t, [955](#)
- pSetIndicationRegReq
  - pack\_sms\_SLQSSetIndicationRegister\_t, [495](#)
- pSetRoutesReq
  - pack\_sms\_SLQSSmsSetRoutes\_t, [497](#)
- pSettingResp
  - unpack\_ims\_SLQSGetIMSSMSConfig\_t, [738](#)
  - unpack\_ims\_SLQSGetIMSUserConfig\_t, [739](#)
  - unpack\_ims\_SLQSGetIMSVoIPConfig\_t, [742](#)
  - unpack\_ims\_SLQSGetRegMgrConfig\_t, [743](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
  - unpack\_ims\_SLQSSetIMSSMSConfig\_t, [746](#)
  - unpack\_ims\_SLQSSetIMSUserConfig\_t, [746](#)
  - unpack\_ims\_SLQSSetIMSVoIPConfig\_t, [747](#)
  - unpack\_ims\_SLQSSetRegMgrConfig\_t, [747](#)
  - unpack\_ims\_SLQSSetSIPConfig\_t, [748](#)
- pSigCompEnabled
  - pack\_ims\_SLQSSetSIPConfig\_t, [406](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pSigIndReq
  - pack\_nas\_SLQSSetSignalStrengthsCallback\_t, [457](#)
- pSignalInfo
  - unpack\_voice\_VoiceInfoRecCallback\_ind\_t, [1047](#)
- pSignalStrengthInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSimBusyStatus
  - unpack\_uim\_GetCardStatusV2\_t, [986](#)
- pSimRejInfo
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [842](#)
- pSmsOnIms
  - sms\_sendAsynsmsParams, [605](#)
- pSmsStorage
  - unpack\_sms\_SLQSSwiGetSMSStorage\_t, [946](#)
- pSourceIP
  - LibPackTFTIDParams, [144](#)
- pSpeedHorizontal
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pSpeedUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pSpeedVertical
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pSpeedVerticalUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
- pSrvDomainPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [473](#)
  - unpack\_nas\_SLQSGetSysSelectionPref\_t, [847](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [850](#)
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pSrvOpt
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, [1021](#)
- pSrvRegRestrict
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pSrvRegRestriction
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [462](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [473](#)
  - unpack\_nas\_SLQSGetSysInfoV2\_t, [842](#)
- pSrvProviderName
  - unpack\_nas\_SLQSGetOperatorNameData\_t, [818](#)
- pStage0Val
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pStage1Val
  - audio\_RXPCMIIRFiltr, [45](#)

- audio\_TXPCMIIRFiltr, [47](#)
- pStage2Val
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pStage3Val
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pStage4Val
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pStageCnt
  - audio\_RXPCMIIRFiltr, [45](#)
  - audio\_TXPCMIIRFiltr, [47](#)
- pStatMask
  - pack\_wds\_GetPacketStatistics\_t, [558](#)
- pSubscribeTimer
  - pack\_ims\_SLQSSetSIPConfig\_t, [407](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pSubscriptionChangeInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSubscriptionInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSupportEmergencyCalls
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pSupportedBitmasks
  - unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t, [967](#)
- pSuppressSysInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSuppsNotifEvents
  - pack\_voice\_SLQSVoiceIndicationRegister\_t, [547](#)
- pSvUsedforFix
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pSvcClass
  - pack\_voice\_SLQSVoiceGetCallBarring\_t, [542](#)
  - pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t, [543](#)
  - pack\_voice\_SLQSVoiceGetCallWaiting\_t, [544](#)
  - unpack\_voice\_SLQSVoiceGetCallBarring\_t, [1017](#)
  - unpack\_voice\_SLQSVoiceGetCallWaiting\_t, [1023](#)
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- pSvcType
  - pack\_voice\_SLQSVoiceDialCall\_t, [541](#)
- pSysInfoInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pSysInfoNoChange
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, [888](#)
- pSysTime
  - unpack\_dms\_GetNetworkTimeV2\_t, [686](#)
- pSystemSelectionInd
  - pack\_nas\_SLQSNasIndicationRegisterExt\_t, [443](#)
- pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pT3402TimerValueInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pTDSCDMABandPref
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [853](#)
- pTDSCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMAECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMAECIOThreshList
  - nas\_TDSCDMAECIOThresh, [330](#)
- pTDSCDMARSCPDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMARSCPTthresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMARSCPTthreshList
  - nas\_TDSCDMARSCPTthresh, [331](#)
- pTDSCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMARSSIThreshList
  - nas\_TDSCDMARSSIThresh, [331](#)
- pTDSCDMASINRDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMASINRThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, [438](#)
- pTDSCDMASINRThreshList
  - nas\_TDSCDMASINRThresh, [332](#)
- pTDSCDMASigInfoExt
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [873](#)
- pTFTID1Params
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pTFTID2Params
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pTTYConfigStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, [1036](#)
- pTTYMode
  - pack\_voice\_SLQSVoiceGetConfig\_t, [546](#)
  - pack\_voice\_SLQSVoiceSetConfig\_t, [551](#)
- pTXAGCList
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pTXAIG
  - audio\_TXAGCList, [45](#)
- pTXAVCSwitch
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pTXComprSlope
  - audio\_TXAGCList, [46](#)
- pTXComprThres

- audio\_TXAGCList, [46](#)
- pTXDroppedCount
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXExpSlope
  - audio\_TXAGCList, [46](#)
- pTXExpThres
  - audio\_TXAGCList, [46](#)
- pTXGain
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pTXInfo
  - unpack\_nas\_SLQSNasGetTxRxInfo\_t, [866](#)
- pTXOKBytesLastCall
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXOKBytesCount
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXPCMIIRFtr
  - pack\_audio\_SLQSSetAudioPathConfig\_t, [376](#)
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, [657](#)
- pTXPacketErrors
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXPacketOverflows
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXPacketSuccesses
  - unpack\_wds\_GetPacketStatistics\_t, [1064](#)
- pTXStaticGain
  - audio\_TXAGCList, [46](#)
- pTXTotalBytes
  - unpack\_wds\_GetByteTotals\_t, [1050](#)
- pTdsdmaBandPref
  - pack\_nas\_SLQSSetSysSelectionPref\_t, [463](#)
  - pack\_nas\_SLQSSetSysSelectionPrefExt\_t, [473](#)
- pTech
  - pack\_wds\_SLQSSetDataSession\_t, [577](#)
- pTechnologyMask
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pTempReport
  - pack\_dms\_SwiSetEventReport\_t, [393](#)
- pTimeSrc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pTimeStamp
  - pack\_pds\_SLQSPDSInjectPositionData\_t, [481](#)
  - packgetDyingGaspStatistics, [582](#)
- pTimeType
  - pack\_pds\_SLQSPDSInjectPositionData\_t, [481](#)
- pTimeUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pTimeZone
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, [870](#)
- pTimeout
  - unpack\_pds\_GetPDSDDefaults\_t, [893](#)
- pTimer
  - pack\_nas\_SLQSNasSwiIndicationRegister\_t, [454](#)
- pTimerExpiryInd
  - pack\_nas\_SLQSNasIndicationRegisterV2\_t, [451](#)
- pTimerSIPReg
  - pack\_ims\_SLQSSetSIPConfig\_t, [407](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pTimerT1
  - pack\_ims\_SLQSSetSIPConfig\_t, [407](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pTimerT2
  - pack\_ims\_SLQSSetSIPConfig\_t, [407](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pTimerTf
  - pack\_ims\_SLQSSetSIPConfig\_t, [407](#)
  - unpack\_ims\_SLQSGetSIPConfig\_t, [744](#)
- pTimerVal
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
- pTimestampUtc
  - unpack\_loc\_BestAvailPos\_Ind\_t, [766](#)
  - unpack\_loc\_PositionRpt\_Ind\_t, [786](#)
- pTrackingStatus
  - unpack\_pds\_GetPDSSState\_t, [894](#)
- pTransLayerInfo
  - sms\_getTransLayerInfo, [598](#)
  - unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t, [947](#)
- pTransferStatInd
  - pack\_wds\_SLQSGetDUNCallInfo\_t, [569](#)
  - pack\_wds\_SLQSWdsSetEventReport\_t, [580](#)
- pTransferStatusReport
  - sms\_setRoutesReq, [607](#)
- pTrueIMSI
  - unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, [856](#)
- pUATI
  - unpack\_nas\_SLQSSwiGetHRPDStats\_t, [881](#)
- pUIMStatusReport
  - pack\_dms\_SwiSetEventReport\_t, [393](#)
- pUMTSCellID
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [857](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pUMTSExtInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pUMTSInfo
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [858](#)
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [859](#)
- pUMTSMInQoS
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pUMTSMInQoSInd
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)
  - LibPackprofile\_3GPP, [135](#)
- pUMTSReqQoS
  - LibpackProfile3GPP, [113](#)
  - LibpackProfile3GPPV2, [129](#)



- LibPackprofile\_3GPP, 135
- pUMTSReqQoSsigInd
  - LibpackProfile3GPP, 113
  - LibpackProfile3GPPV2, 129
  - LibPackprofile\_3GPP, 135
- pURL
  - unpack\_loc\_GetServer\_Ind\_t, 775
- pURLAddr
  - pack\_loc\_SLQSLOCSetServer\_t, 422
- pUSSDInfo
  - unpack\_voice\_SLQSOriginateUSSD\_t, 1009
- pUSSInfo
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1042
- pUUSInfo
  - pack\_voice\_SLQSVoiceDialCall\_t, 541
- pUUSInfo
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1022
- pUimAutoSwitchActSlot
  - unpack\_dms\_SLQSDmsSwiGetUimSelection\_t, 706
- pUimSlotsStatus
  - unpack\_uim\_SLQSUIMGetSlotsStatus\_t, 994
- pUmtsRoamPDPTType
  - LibpackProfile3GPPV2, 129
- pUname
  - PackSwiAVMSSettingsAPNInfo, 585
- pUsageSetting
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 854
- pUsbMTUSize
  - unpack\_swidms\_SLQSSwiDmsGetMTU\_t, 965
- pUser
  - pack\_wds\_SLQSStartDataSession\_t, 577
- pUserConfigEvents
  - pack\_ims\_SLQSImsConfigIndicationRegister\_t, 401
- pUserData
  - sms\_sendAsynsmsParams, 605
- pUserId
  - LibpackProfile3GPP2, 119
  - LibPackprofile\_3GPP2, 140
- pUserIdSize
  - LibpackProfile3GPP2, 119
  - LibPackprofile\_3GPP2, 140
- pUsername
  - LibpackProfile3GPP, 113
  - LibpackProfile3GPPV2, 129
  - LibPackprofile\_3GPP, 135
  - pack\_wds\_SetDefaultProfile\_t, 562
- pUsernameSize
  - LibpackProfile3GPP, 113
  - LibpackProfile3GPPV2, 129
  - LibPackprofile\_3GPP, 135
- pUstTime
  - unpack\_dms\_GetNetworkTimeV2\_t, 686
- pValidCardStatus
  - unpack\_uim\_GetCardStatusV2\_t, 986
- pVerboseFailReasonType
  - unpack\_wds\_SLQSStartDataSession\_t, 1096
- pVerboseFailureReason
  - unpack\_wds\_SLQSStartDataSession\_t, 1096
- pVertConfidence
  - unpack\_loc\_BestAvailPos\_Ind\_t, 766
  - unpack\_loc\_PositionRpt\_Ind\_t, 787
- pVertReliability
  - unpack\_loc\_BestAvailPos\_Ind\_t, 767
  - unpack\_loc\_PositionRpt\_Ind\_t, 787
- pVertUnc
  - unpack\_loc\_BestAvailPos\_Ind\_t, 767
  - unpack\_loc\_PositionRpt\_Ind\_t, 787
- pVerticalConfidence
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pVerticalUnc
  - pack\_pds\_SLQSPDSInjectPositionData\_t, 481
- pVoIPConfigEvents
  - pack\_ims\_SLQSImsConfigIndicationRegister\_t, 401
- pVoiceDomainPref
  - pack\_voice\_SLQSVoiceGetConfig\_t, 546
  - unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 854
- pVoiceDomainPrefStatus
  - unpack\_voice\_SLQSVoiceSetConfig\_t, 1036
- pVoicePrivacy
  - unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
  - unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1022
- pVoltReport
  - pack\_dms\_SwiSetEventReport\_t, 393
- pVolume
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, 501
- pWAMSParaChanged
  - unpack\_swiaavms\_SLQSAVMSEventReportInd\_t, 955
- pWCDMABER
  - unpack\_nas\_SLQSGetErrorRate\_t, 814
- pWCDMACallBarringSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pWCDMACellInfoExt
  - unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, 860
- pWCDMACipherDomainSysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pWCDMAECIODelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pWCDMAECIOThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pWCDMAECIOThreshList
  - nas\_WCDMAECIOThresh, 346
- pWCDMAInfoLTENeighborCell
  - unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 858

- unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, 860
- pWCDMARSSIDelta
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pWCDMARSSIThresh
  - pack\_nas\_SLQSNasConfigSigInfo2\_t, 438
- pWCDMARSSIThreshList
  - nas\_WCDMARSSIThresh, 348
- pWCDMASigInfo
  - unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 873
- pWCDMASrvStatusInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pWCDMASysInfo
  - unpack\_nas\_SLQSGetSysInfo\_t, 837
  - unpack\_nas\_SLQSGetSysInfoV2\_t, 842
  - unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- pWcdmaUARFCN
  - pack\_nas\_SLQSNASSwiSetChannelLock\_t, 455
  - unpack\_nas\_SLQSNASSwiGetChannelLock\_t, 873
- pWifiState
  - pack\_pds\_SLQSSetPositionMethodState\_t, 483
- pXid
  - unpack\_loc\_BestAvailPos\_Ind\_t, 767
- pXtraDataState
  - pack\_pds\_SLQSSetPositionMethodState\_t, 483
- pXtraTimeState
  - pack\_pds\_SLQSSetPositionMethodState\_t, 483
- pack\_audio\_SLQSGetAudioPathConfig
  - audio.h, 1184
- pack\_audio\_SLQSGetAudioPathConfig\_t, 372
  - Item, 373
  - Profile, 373
- pack\_audio\_SLQSGetAudioProfile
  - audio.h, 1184
- pack\_audio\_SLQSGetAudioProfile\_t, 373
  - Generator, 374
- pack\_audio\_SLQSGetAudioVolTLBConfig
  - audio.h, 1185
- pack\_audio\_SLQSGetAudioVolTLBConfig\_t, 374
  - Generator, 374
  - Item, 374
  - Profile, 374
  - Volume, 374
- pack\_audio\_SLQSSetAudioPathConfig
  - audio.h, 1185
- pack\_audio\_SLQSSetAudioPathConfig\_t, 374
  - pCodecSTGain, 376
  - pECMode, 376
  - pNSEnable, 376
  - pRXAGCList, 376
  - pRXAVCList, 376
  - pTXAGCList, 376
  - pTXAVCSwitch, 376
  - pTXGain, 376
  - Profile, 376
- pack\_audio\_SLQSSetAudioProfile
  - audio.h, 1186
- pack\_audio\_SLQSSetAudioProfile\_t, 376
  - EarMute, 377
  - Generator, 377
  - MicMute, 377
  - Profile, 377
  - Volume, 377
- pack\_audio\_SLQSSetAudioVolTLBConfig
  - audio.h, 1186
- pack\_audio\_SLQSSetAudioVolTLBConfig\_t, 378
  - Generator, 378
  - Item, 378
  - Profile, 378
  - VolValue, 378
  - Volume, 378
- pack\_cat\_CATSendEnvelopeCommand
  - cat.h, 1190
- pack\_cat\_CATSendEnvelopeCommand\_t, 378
  - cmdID, 379
  - dataLen, 379
  - pData, 379
- pack\_cat\_CATSendTerminalResponse
  - cat.h, 1190
- pack\_cat\_CATSendTerminalResponse\_t, 379
  - dataLen, 380
  - pData, 380
  - refID, 380
- pack\_cat\_SetCATEventCallback
  - cat.h, 1190
- pack\_cat\_SetCATEventCallback\_t, 380
  - eventMask, 380
- pack\_dms\_ActivateAutomatic
  - dms.h, 1208
- pack\_dms\_ActivateAutomatic\_t, 381
  - actCode, 381
- pack\_dms\_GetActivationState
  - dms.h, 1208
- pack\_dms\_GetBandCapability
  - dms.h, 1208
- pack\_dms\_GetCrashAction
  - dms.h, 1209
- pack\_dms\_GetCustFeature
  - dms.h, 1209
- pack\_dms\_GetCustFeaturesV2
  - dms.h, 1209
- pack\_dms\_GetCustFeaturesV2\_t, 381
  - cust\_id, 381
  - list\_type, 381
  - Tlvresult, 381
- pack\_dms\_GetDeviceCap
  - dms.h, 1210
- pack\_dms\_GetDeviceCapabilities
  - dms.h, 1210
- pack\_dms\_GetDeviceCapabilitiesV2
  - dms.h, 1211
- pack\_dms\_GetDeviceHardwareRev
  - dms.h, 1211

- pack\_dms\_GetDeviceMfr  
dms.h, [1211](#)
- pack\_dms\_GetDeviceSerialNumbers  
dms.h, [1212](#)
- pack\_dms\_GetFSN  
dms.h, [1213](#)
- pack\_dms\_GetFirmwareInfo  
dms.h, [1212](#)
- pack\_dms\_GetFirmwareRevision  
dms.h, [1212](#)
- pack\_dms\_GetFirmwareRevisions  
dms.h, [1213](#)
- pack\_dms\_GetHardwareRevision  
dms.h, [1213](#)
- pack\_dms\_GetIMSI  
dms.h, [1214](#)
- pack\_dms\_GetManufacturer  
dms.h, [1214](#)
- pack\_dms\_GetModelID  
dms.h, [1214](#)
- pack\_dms\_GetNetworkTime  
dms.h, [1215](#)
- pack\_dms\_GetNetworkTimeV2  
dms.h, [1215](#)
- pack\_dms\_GetOfflineReason  
dms.h, [1215](#)
- pack\_dms\_GetPRLVersion  
dms.h, [1216](#)
- pack\_dms\_GetPower  
dms.h, [1216](#)
- pack\_dms\_GetSerialNumbers  
dms.h, [1216](#)
- pack\_dms\_GetUSBComp  
dms.h, [1217](#)
- pack\_dms\_GetVoiceNumber  
dms.h, [1217](#)
- pack\_dms\_ResetToFactoryDefaults  
dms.h, [1218](#)
- pack\_dms\_ResetToFactoryDefaults\_t, [382](#)  
spc, [382](#)
- pack\_dms\_SLQSDmsSwiGetPCInfo  
dms.h, [1221](#)
- pack\_dms\_SLQSDmsSwiGetResetInfo  
dms.h, [1222](#)
- pack\_dms\_SLQSDmsSwiGetUimSelection  
dms.h, [1222](#)
- pack\_dms\_SLQSDmsSwiIndicationRegister  
dms.h, [1222](#)
- pack\_dms\_SLQSDmsSwiIndicationRegister\_t, [389](#)  
resetInfoInd, [389](#)
- pack\_dms\_SLQSGetBandCapability  
dms.h, [1223](#)
- pack\_dms\_SLQSGetERIFile  
dms.h, [1223](#)
- pack\_dms\_SLQSGetPowerSaveModeConfig  
dms.h, [1223](#)
- pack\_dms\_SLQSSetPowerSaveModeConfig  
dms.h, [1224](#)
- pack\_dms\_SLQSSetPowerSaveModeConfig\_t, [389](#)  
pActiveTimer, [390](#)  
pPeriodicUpdateTimer, [390](#)  
pPsmEnableState, [390](#)
- pack\_dms\_SLQSSwiClearDyingGaspStatistics  
dms.h, [1224](#)
- pack\_dms\_SLQSSwiGetCrashInfo  
dms.h, [1225](#)
- pack\_dms\_SLQSSwiGetCrashInfo\_t, [390](#)  
clear, [390](#)
- pack\_dms\_SLQSSwiGetDyingGaspCfg  
dms.h, [1225](#)
- pack\_dms\_SLQSSwiGetDyingGaspStatistics  
dms.h, [1225](#)
- pack\_dms\_SLQSSwiGetFirmwareCurr  
dms.h, [1226](#)
- pack\_dms\_SLQSSwiGetFwUpdateStatus  
dms.h, [1226](#)
- pack\_dms\_SLQSSwiGetHostDevInfo  
dms.h, [1226](#)
- pack\_dms\_SLQSSwiGetOSInfo  
dms.h, [1227](#)
- pack\_dms\_SLQSSwiGetSerialNoExt  
dms.h, [1227](#)
- pack\_dms\_SLQSSwiSetDyingGaspCfg  
dms.h, [1227](#)
- pack\_dms\_SLQSSwiSetDyingGaspCfg\_t, [390](#)  
pDestSMSContent, [391](#)  
pDestSMSNum, [391](#)
- pack\_dms\_SLQSSwiSetHostDevInfo  
dms.h, [1228](#)
- pack\_dms\_SLQSSwiSetHostDevInfo\_t, [391](#)  
hostID, [392](#)  
manString, [392](#)  
modelString, [392](#)  
plasmaIDString, [392](#)  
swVerString, [392](#)
- pack\_dms\_SLQSSwiSetOSInfo  
dms.h, [1228](#)
- pack\_dms\_SLQSSwiSetOSInfo\_t, [392](#)  
nameString, [392](#)  
versionString, [392](#)
- pack\_dms\_SLQSUIGetState  
dms.h, [1228](#)
- pack\_dms\_SetActivationStatusCallback  
dms.h, [1218](#)
- pack\_dms\_SetActivationStatusCallback\_t, [382](#)  
activationState, [382](#)
- pack\_dms\_SetCrashAction  
dms.h, [1218](#)
- pack\_dms\_SetCrashAction\_t, [382](#)  
crashAction, [383](#)
- pack\_dms\_SetCustFeature  
dms.h, [1219](#)
- pack\_dms\_SetCustFeature\_t, [383](#)  
DHCPRelayEnabled, [385](#)  
DisableIMSI, [385](#)  
GPSPMP, [385](#)



- GPSSel, [385](#)
- GpsEnable, [385](#)
- IPFamSupport, [385](#)
- IsVoiceEnabled, [385](#)
- RMAutoConnect, [385](#)
- SMSSupport, [385](#)
- pack\_dms\_SetCustFeaturesV2
  - dms.h, [1219](#)
- pack\_dms\_SetCustFeaturesV2\_t, [385](#)
  - cust\_id, [386](#)
  - cust\_value, [386](#)
  - Tlvresult, [386](#)
  - value\_length, [386](#)
- pack\_dms\_SetEventReport
  - dms.h, [1219](#)
- pack\_dms\_SetEventReport\_t, [386](#)
  - mode, [386](#)
- pack\_dms\_SetFirmwarePreference
  - dms.h, [1220](#)
- pack\_dms\_SetIndicationRegister
  - dms.h, [1220](#)
- pack\_dms\_SetIndicationRegister\_t, [386](#)
  - PSMCfgChangeInfo, [387](#)
  - PSMStatus, [387](#)
  - RptIMSCapability, [387](#)
- pack\_dms\_SetPower
  - dms.h, [1220](#)
- pack\_dms\_SetPower\_t, [387](#)
  - mode, [388](#)
  - Tlvresult, [388](#)
- pack\_dms\_SetUSBComp
  - dms.h, [1221](#)
- pack\_dms\_SetUSBComp\_t, [388](#)
  - Tlvresult, [388](#)
  - USBComp, [388](#)
- pack\_dms\_SwiSetEventReport
  - dms.h, [1229](#)
- pack\_dms\_SwiSetEventReport\_t, [392](#)
  - pTempReport, [393](#)
  - pUIMStatusReport, [393](#)
  - pVoltReport, [393](#)
- pack\_dms\_SwiUimSelect
  - dms.h, [1229](#)
- pack\_dms\_SwiUimSelect\_t, [393](#)
  - uim\_select, [393](#)
- pack\_dms\_UIMChangePIN
  - dms.h, [1229](#)
- pack\_dms\_UIMChangePIN\_t, [394](#)
  - id, [394](#)
  - newValue, [394](#)
  - oldValue, [394](#)
- pack\_dms\_UIMGetControlKeyStatus
  - dms.h, [1230](#)
- pack\_dms\_UIMGetControlKeyStatus\_t, [394](#)
  - facility, [395](#)
- pack\_dms\_UIMGetICCID
  - dms.h, [1230](#)
- pack\_dms\_UIMGetICCID\_t, [395](#)
  - ParamPresenceMask, [395](#)
  - Tlvresult, [395](#)
- pack\_dms\_UIMGetPINStatus
  - dms.h, [1231](#)
- pack\_dms\_UIMSetControlKeyProtection
  - dms.h, [1231](#)
- pack\_dms\_UIMSetControlKeyProtection\_t, [395](#)
  - facility, [396](#)
  - facilityCk, [396](#)
  - facilityState, [396](#)
- pack\_dms\_UIMSetPINProtection
  - dms.h, [1231](#)
- pack\_dms\_UIMSetPINProtection\_t, [396](#)
  - bEnable, [397](#)
  - id, [397](#)
  - value, [397](#)
- pack\_dms\_UIMUnblockControlKey
  - dms.h, [1232](#)
- pack\_dms\_UIMUnblockControlKey\_t, [397](#)
  - facility, [397](#)
  - facilityCk, [397](#)
- pack\_dms\_UIMUnblockPIN
  - dms.h, [1232](#)
- pack\_dms\_UIMUnblockPIN\_t, [397](#)
  - id, [398](#)
  - newPin, [398](#)
  - pukValue, [398](#)
- pack\_dms\_UIMVerifyPIN
  - dms.h, [1232](#)
- pack\_dms\_UIMVerifyPIN\_t, [398](#)
  - id, [398](#)
  - value, [398](#)
- pack\_dms\_ValidateSPC
  - dms.h, [1233](#)
- pack\_fms\_GetImagesPreference
  - fms.h, [1261](#)
- pack\_fms\_GetImagesPreference\_t, [398](#)
  - Tlvresult, [399](#)
- pack\_fms\_GetStoredImages
  - fms.h, [1262](#)
- pack\_fms\_GetStoredImages\_t, [399](#)
  - Tlvresult, [399](#)
- pack\_fms\_SetImagesPreference
  - fms.h, [1262](#)
- pack\_fms\_SetImagesPreference\_t, [399](#)
  - bForceDownload, [400](#)
  - imageListSize, [400](#)
  - modemindex, [400](#)
  - pImageList, [400](#)
  - Tlvresult, [400](#)
- pack\_ims\_SLQSGetIMSSMSConfig
  - ims.h, [1265](#)
- pack\_ims\_SLQSGetIMSUserConfig
  - ims.h, [1265](#)
- pack\_ims\_SLQSGetIMSVoIPConfig
  - ims.h, [1266](#)
- pack\_ims\_SLQSGetRegMgrConfig
  - ims.h, [1266](#)

- pack\_ims\_SLQSGetSIPConfig
  - ims.h, [1267](#)
- pack\_ims\_SLQSImsConfigIndicationRegister
  - ims.h, [1267](#)
- pack\_ims\_SLQSImsConfigIndicationRegister\_t, [400](#)
  - pRegMgrConfigEvents, [401](#)
  - pSIPConfigEvents, [401](#)
  - pSMSConfigEvents, [401](#)
  - pUserConfigEvents, [401](#)
  - pVoIPConfigEvents, [401](#)
- pack\_ims\_SLQSSetIMSSMSConfig
  - ims.h, [1268](#)
- pack\_ims\_SLQSSetIMSSMSConfig\_t, [401](#)
- pack\_ims\_SLQSSetIMSUserConfig
  - ims.h, [1268](#)
- pack\_ims\_SLQSSetIMSUserConfig\_t, [402](#)
  - pIMSDomain, [403](#)
- pack\_ims\_SLQSSetIMSVoIPConfig
  - ims.h, [1269](#)
- pack\_ims\_SLQSSetIMSVoIPConfig\_t, [403](#)
  - pAmrMode, [405](#)
  - pRingingTimer, [405](#)
- pack\_ims\_SLQSSetRegMgrConfig
  - ims.h, [1269](#)
- pack\_ims\_SLQSSetRegMgrConfig\_t, [405](#)
  - pCSCFPortName, [406](#)
  - pIMSTestMode, [406](#)
  - pPriCSCFPort, [406](#)
- pack\_ims\_SLQSSetSIPConfig
  - ims.h, [1269](#)
- pack\_ims\_SLQSSetSIPConfig\_t, [406](#)
  - pSIPLocalPort, [407](#)
  - pSigCompEnabled, [406](#)
  - pSubscribeTimer, [407](#)
  - pTimerSIPReg, [407](#)
  - pTimerT1, [407](#)
  - pTimerT2, [407](#)
  - pTimerTf, [407](#)
- pack\_imsa\_SLQSGetIMSARegStatus
  - imsa.h, [1277](#)
- pack\_imsa\_SLQSGetIMSAServiceStatus
  - imsa.h, [1277](#)
- pack\_imsa\_SLQSRegisterIMSARegistration
  - imsa.h, [1278](#)
- pack\_imsa\_SLQSRegisterIMSARegistration\_t, [407](#)
  - PdpStatusConfig, [408](#)
  - RatHandoverStatusConfig, [408](#)
  - RegStatusConfig, [408](#)
  - ServiceStatusConfig, [408](#)
- pack\_loc\_Delete\_Assist\_Data\_t, [408](#)
  - pBdsSVInfo, [409](#)
  - pCellDb, [409](#)
  - pClkInfo, [409](#)
  - pGnssData, [409](#)
  - pSVInfo, [409](#)
  - Tlvresult, [409](#)
- pack\_loc\_DeleteAssistData
  - loc.h, [1301](#)
- pack\_loc\_EventRegister
  - loc.h, [1302](#)
- pack\_loc\_EventRegister\_t, [409](#)
  - eventRegister, [411](#)
  - Tlvresult, [411](#)
- pack\_loc\_GetFixCriteria
  - loc.h, [1302](#)
- pack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, [1303](#)
- pack\_loc\_SLQSLOCGetBestAvailPos\_t, [412](#)
  - Tlvresult, [412](#)
  - xid, [412](#)
- pack\_loc\_SLQSLOCGetOpMode
  - loc.h, [1304](#)
- pack\_loc\_SLQSLOCGetServer
  - loc.h, [1304](#)
- pack\_loc\_SLQSLOCGetServer\_t, [413](#)
  - has\_serverAddrTypeMask, [413](#)
  - serverAddrTypeMask, [413](#)
  - serverType, [413](#)
- pack\_loc\_SLQSLOCInjectPosition
  - loc.h, [1304](#)
- pack\_loc\_SLQSLOCInjectPosition\_t, [413](#)
  - altitudeSrcInfo, [417](#)
  - altitudeWrtEllipsoid, [417](#)
  - altitudeWrtMeanSeaLevel, [417](#)
  - has\_altitudeSrcInfo, [417](#)
  - has\_altitudeWrtEllipsoid, [417](#)
  - has\_horConfidence, [417](#)
  - has\_horReliability, [417](#)
  - has\_horUncCircular, [417](#)
  - has\_latitude, [417](#)
  - has\_longitude, [417](#)
  - has\_positionSrc, [417](#)
  - has\_rawHorConfidence, [417](#)
  - has\_rawHorUncCircular, [417](#)
  - has\_timestampAge, [417](#)
  - has\_timestampUtc, [417](#)
  - has\_vertConfidence, [417](#)
  - has\_vertReliability, [417](#)
  - has\_vertUnc, [418](#)
  - horConfidence, [418](#)
  - horReliability, [418](#)
  - horUncCircular, [418](#)
  - latitude, [418](#)
  - longitude, [418](#)
  - positionSrc, [418](#)
  - rawHorConfidence, [418](#)
  - rawHorUncCircular, [418](#)
  - timestampAge, [418](#)
  - timestampUtc, [418](#)
  - vertConfidence, [418](#)
  - vertReliability, [418](#)
  - vertUnc, [418](#)
- pack\_loc\_SLQSLOCInjectSensorData
  - loc.h, [1305](#)
- pack\_loc\_SLQSLOCInjectSensorData\_t, [418](#)
  - accelTemp, [419](#)

- acceleroData, [419](#)
- acceleroTimeSrc, [419](#)
- gyroData, [419](#)
- gyroTemp, [419](#)
- gyroTimeSrc, [420](#)
- has\_accelTemp, [420](#)
- has\_acceleroTimeSrc, [420](#)
- has\_accleroData, [420](#)
- has\_gyroData, [420](#)
- has\_gyroTemp, [420](#)
- has\_gyroTimeSrc, [420](#)
- has\_opaqueId, [420](#)
- opaqueId, [420](#)
- pack\_loc\_SLQSLOCInjectUTCTime
  - loc.h, [1305](#)
- pack\_loc\_SLQSLOCInjectUTCTime\_t, [420](#)
  - timeMsec, [420](#)
  - timeUncMsec, [420](#)
- pack\_loc\_SLQSLOCSetCradleMountConfig
  - loc.h, [1306](#)
- pack\_loc\_SLQSLOCSetCradleMountConfig\_t, [420](#)
  - confidence, [421](#)
  - has\_confidence, [421](#)
  - state, [421](#)
- pack\_loc\_SLQSLOCSetServer
  - loc.h, [1306](#)
- pack\_loc\_SLQSLOCSetServer\_t, [421](#)
  - pIPv4Config, [422](#)
  - pIPv6Config, [422](#)
  - pURLAddr, [422](#)
  - serverType, [422](#)
- pack\_loc\_SetExtPowerState
  - loc.h, [1302](#)
- pack\_loc\_SetExtPowerState\_t, [411](#)
  - extPowerState, [411](#)
  - Tlvresult, [411](#)
- pack\_loc\_SetOperationMode
  - loc.h, [1303](#)
- pack\_loc\_SetOperationMode\_t, [411](#)
  - mode, [412](#)
  - Tlvresult, [412](#)
- pack\_loc\_Start
  - loc.h, [1306](#)
- pack\_loc\_Start\_t, [422](#)
  - pApplicationInfo, [423](#)
  - pConfigAltitudeAssumed, [423](#)
  - pHorizontalAccuracyLvl, [423](#)
  - pIntermediateReportState, [423](#)
  - pMinIntervalTime, [423](#)
  - pRecurrenceType, [423](#)
  - SessionId, [424](#)
  - Tlvresult, [424](#)
- pack\_loc\_Stop
  - loc.h, [1307](#)
- pack\_loc\_Stop\_t, [424](#)
  - SessionId, [424](#)
  - Tlvresult, [424](#)
- pack\_nas\_GetACCOLC
  - nas.h, [1335](#)
- pack\_nas\_GetANAAAAAuthenticationStatus
  - nas.h, [1335](#)
- pack\_nas\_GetCDMANetworkParameters
  - nas.h, [1336](#)
- pack\_nas\_GetHomeNetwork
  - nas.h, [1336](#)
- pack\_nas\_GetHomeNetwork3GPP2
  - nas.h, [1336](#)
- pack\_nas\_GetNetworkPreference
  - nas.h, [1337](#)
- pack\_nas\_GetRFInfo
  - nas.h, [1337](#)
- pack\_nas\_GetServingNetwork
  - nas.h, [1337](#)
- pack\_nas\_GetServingNetworkCapabilities
  - nas.h, [1337](#)
- pack\_nas\_GetSignalStrengths
  - nas.h, [1338](#)
- pack\_nas\_InitiateDomainAttach
  - nas.h, [1338](#)
- pack\_nas\_InitiateDomainAttach\_t, [424](#)
  - action, [424](#)
- pack\_nas\_PerformNetworkScan
  - nas.h, [1338](#)
- pack\_nas\_PerformNetworkScanPCI
  - nas.h, [1339](#)
- pack\_nas\_PerformNetworkScanPCI\_t, [425](#)
  - pCiotOpModePref, [427](#)
  - pLteM1BandPref, [427](#)
  - pLteNB1BandPref, [428](#)
  - pNetworkType, [428](#)
  - pScanType, [428](#)
- pack\_nas\_SLQSConfigSigInfo
  - nas.h, [1341](#)
- pack\_nas\_SLQSConfigSigInfo\_t, [431](#)
  - pECIOThresh, [432](#)
  - pLOThresh, [432](#)
  - pLTESNRThresh, [432](#)
  - pLTESigRptCfg, [432](#)
  - pRSRPThresh, [432](#)
  - pRSRQThresh, [432](#)
  - pRSSIThresh, [432](#)
- pack\_nas\_SLQSGetErrorRate
  - nas.h, [1341](#)
- pack\_nas\_SLQSGetHomeNetwork
  - nas.h, [1342](#)
- pack\_nas\_SLQSGetNetworkTime
  - nas.h, [1342](#)
- pack\_nas\_SLQSGetOperatorNameData
  - nas.h, [1342](#)
- pack\_nas\_SLQSGetPLMNName
  - nas.h, [1343](#)
- pack\_nas\_SLQSGetPLMNName\_t, [432](#)
  - mcc, [433](#)
  - mnc, [433](#)
  - pMncPcsStatus, [433](#)
- pack\_nas\_SLQSGetServingSystem

- nas.h, [1343](#)
- pack\_nas\_SLQSGetservingSystemV2
  - nas.h, [1343](#)
- pack\_nas\_SLQSGetSignalStrength
  - nas.h, [1344](#)
- pack\_nas\_SLQSGetSysInfo
  - nas.h, [1344](#)
- pack\_nas\_SLQSGetSysInfoV2
  - nas.h, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPref
  - nas.h, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPrefExt
  - nas.h, [1345](#)
- pack\_nas\_SLQSGetSysSelectionPrefExtV2
  - nas.h, [1346](#)
- pack\_nas\_SLQSIInitiateNetworkRegistration
  - nas.h, [1346](#)
- pack\_nas\_SLQSIInitiateNetworkRegistration\_t, [433](#)
  - pChangeDuration, [434](#)
  - pMNRInfo, [434](#)
  - pMncPcsDigitStatus, [434](#)
  - regAction, [434](#)
- pack\_nas\_SLQSNASGetForbiddenNetworks
  - nas.h, [1348](#)
- pack\_nas\_SLQSNASGetDRXParams
  - nas.h, [1348](#)
- pack\_nas\_SLQSNASGetDRXParamsExt
  - nas.h, [1348](#)
- pack\_nas\_SLQSNASGetDRXParamsExt\_t, [439](#)
- pack\_nas\_SLQSNASSetDRXParams
  - nas.h, [1351](#)
- pack\_nas\_SLQSNASSetDRXParams\_t, [451](#)
  - pCycleLen, [452](#)
  - pEdrxEnable, [453](#)
- pack\_nas\_SLQSNASSwiGetChannelLock
  - nas.h, [1351](#)
- pack\_nas\_SLQSNASSwiSetChannelLock
  - nas.h, [1352](#)
- pack\_nas\_SLQSNASSwiSetChannelLock\_t, [454](#)
- pack\_nas\_SLQSNasConfigSigInfo2
  - nas.h, [1346](#)
- pack\_nas\_SLQSNasConfigSigInfo2\_t, [434](#)
  - pHDRIODelta, [438](#)
  - pHDRIOTresh, [438](#)
  - pLTESigRptConfig, [438](#)
- pack\_nas\_SLQSNasGet3GPP2Subscription
  - nas.h, [1347](#)
- pack\_nas\_SLQSNasGet3GPP2Subscription\_t, [439](#)
  - namID, [439](#)
- pack\_nas\_SLQSNasGetCellLocationInfo
  - nas.h, [1347](#)
- pack\_nas\_SLQSNasGetCellLocationInfoV2
  - nas.h, [1347](#)
- pack\_nas\_SLQSNasGetHDRColorCode
  - nas.h, [1349](#)
- pack\_nas\_SLQSNasGetRFInfo
  - nas.h, [1349](#)
- pack\_nas\_SLQSNasGetSigInfo
  - nas.h, [1349](#)
- pack\_nas\_SLQSNasGetTxRxInfo
  - nas.h, [1350](#)
- pack\_nas\_SLQSNasGetTxRxInfo\_t, [440](#)
  - radio\_if, [440](#)
- pack\_nas\_SLQSNasIndicationRegisterExt
  - nas.h, [1350](#)
- pack\_nas\_SLQSNasIndicationRegisterExt\_t, [440](#)
  - pDDTMInd, [443](#)
  - pDualStandByPrefInd, [443](#)
  - pEdrxChangeInfoInd, [443](#)
  - pErrorRateInd, [443](#)
  - pHDRSessionCloseInd, [443](#)
  - pLTECphyCa, [443](#)
  - pManagedRoamingInd, [443](#)
  - pNetworkRejectInd, [443](#)
  - pNetworkTimeInd, [443](#)
  - pServingSystemInd, [443](#)
  - pSignalStrengthInd, [443](#)
  - pSubscriptionInfoInd, [443](#)
  - pSuppressSysInfoInd, [443](#)
  - pSysInfoInd, [443](#)
  - pSystemSelectionInd, [443](#)
- pack\_nas\_SLQSNasIndicationRegisterV2
  - nas.h, [1350](#)
- pack\_nas\_SLQSNasIndicationRegisterV2\_t, [443](#)
  - pAccessClassBarringInd, [450](#)
  - pCallModeStatusInd, [450](#)
  - pDDTMInd, [450](#)
  - pDataSubscriptionPriorityInd, [450](#)
  - pDualStandByPrefInd, [450](#)
  - pE911StateReadyStatusInd, [450](#)
  - pEdrxChangeInfoInd, [450](#)
  - pEmergencyModeStatusInd, [450](#)
  - pErrorRateInd, [450](#)
  - pGcellInfoInd, [450](#)
  - pHDRSessionCloseInd, [450](#)
  - pIMSPrefStatusInd, [450](#)
  - pLTECphyCa, [450](#)
  - pManagedRoamingInd, [451](#)
  - pNetworkRejectInd, [451](#)
  - pNetworkTimeInd, [451](#)
  - pOperatorNameDataInd, [451](#)
  - pRFBandInfoInd, [451](#)
  - pRTREConfigurationInd, [451](#)
  - pSSACBarringInd, [451](#)
  - pServingSystemInd, [451](#)
  - pSignalStrengthInd, [451](#)
  - pSubscriptionChangeInd, [451](#)
  - pSubscriptionInfoInd, [451](#)
  - pSuppressSysInfoInd, [451](#)
  - pSysInfoInd, [451](#)
  - pSystemSelectionInd, [451](#)
  - pT3346TimerStatusChangeInd, [451](#)
  - pT3402TimerValueInd, [451](#)
  - pTimerExpiryInd, [451](#)
- pack\_nas\_SLQSNasSwiIndicationRegister
  - nas.h, [1352](#)

- pack\_nas\_SLQSNasSwtIndicationRegister\_t, 453
  - gsmUmtsDI, 454
  - gsmUmtsUI, 454
  - lteEmmDI, 454
  - lteEmmUI, 454
  - lteEsmDI, 454
  - lteEsmUI, 454
  - pRankIndicatorInd, 454
  - pTimer, 454
- pack\_nas\_SLQSNasSwtModemStatus
  - nas.h, 1352
- pack\_nas\_SLQSPerformNetworkScanV2
  - nas.h, 1353
- pack\_nas\_SLQSSetBandPreference
  - nas.h, 1353
- pack\_nas\_SLQSSetBandPreference\_t, 455
  - bandPref, 456
- pack\_nas\_SLQSSetSignalStrengthsCallback
  - nas.h, 1353
- pack\_nas\_SLQSSetSignalStrengthsCallback\_t, 457
  - bEnable, 457
  - pSigIndReq, 457
- pack\_nas\_SLQSSetSysSelectionPref
  - nas.h, 1354
- pack\_nas\_SLQSSetSysSelectionPref\_t, 457
  - pAcqOrderPref, 462
  - pBandPref, 462
  - pCSGID, 462
  - pChgDuration, 462
  - pEmerMode, 462
  - pGWAqOrderPref, 462
  - pLTEBandPref, 462
  - pModePref, 462
  - pNetSelPref, 462
  - pPRLPref, 462
  - pRAT, 462
  - pRoamPref, 462
  - pSrvDomainPref, 462
  - pSrvRegRestriction, 462
  - pTdsdmaBandPref, 463
- pack\_nas\_SLQSSetSysSelectionPrefExt
  - nas.h, 1354
- pack\_nas\_SLQSSetSysSelectionPrefExt\_t, 463
  - pAcqOrderPref, 472
  - pBandPref, 472
  - pCSGID, 472
  - pChgDuration, 472
  - pCiotAcqOrderPref, 472
  - pCiotLteOpMode, 472
  - pCiotLteOpModePref, 472
  - pEmerMode, 472
  - pGWAqOrderPref, 472
  - pLTEBandPref, 472
  - pLteM1BandPref, 472
  - pLteNb1BandPref, 472
  - pModePref, 472
  - pNetSelPref, 472
  - pNr5gBandPref, 472
  - pPRLPref, 472
  - pRAT, 473
  - pRoamPref, 473
  - pSrvDomainPref, 473
  - pSrvRegRestriction, 473
  - pTdsdmaBandPref, 473
- pack\_nas\_SLQSSwiGetHDRPersonality
  - nas.h, 1355
- pack\_nas\_SLQSSwiGetHDRProtSubtype
  - nas.h, 1355
- pack\_nas\_SLQSSwiGetHRPDStats
  - nas.h, 1355
- pack\_nas\_SLQSSwiGetLteCQI
  - nas.h, 1356
- pack\_nas\_SLQSSwiGetLteSccRxInfo
  - nas.h, 1356
- pack\_nas\_SLQSSwiNetworkDebug
  - nas.h, 1356
- pack\_nas\_SLQSSwiPSDetach
  - nas.h, 1357
- pack\_nas\_SLQSSwiPSDetach\_t, 473
  - pDetachAction, 473
- pack\_nas\_SetACCOLC
  - nas.h, 1339
- pack\_nas\_SetACCOLC\_t, 428
  - accolc, 428
  - spc, 428
- pack\_nas\_SetCDMANetworkParameters
  - nas.h, 1340
- pack\_nas\_SetCDMANetworkParameters\_t, 428
  - pApplication, 429
  - pBroadcast, 429
  - pCustomSCP, 429
  - pForceRev0, 430
  - pProtocol, 430
  - pRoaming, 430
  - pSPC, 430
- pack\_nas\_SetLURRejectCallback
  - nas.h, 1340
- pack\_nas\_SetNetworkPreference
  - nas.h, 1340
- pack\_nas\_SetNetworkPreference\_t, 430
  - Duration, 430
  - TechnologyPref, 430
  - Tlvresult, 431
- pack\_nas\_SetRFInfoCallback
  - nas.h, 1341
- pack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, 1342
- pack\_pds\_ForceXTRADownload
  - pds.h, 1390
- pack\_pds\_GetPDSDDefaults
  - pds.h, 1391
- pack\_pds\_GetPDSSState
  - pds.h, 1391
- pack\_pds\_GetPortAutomaticTracking
  - pds.h, 1391
- pack\_pds\_GetServiceAutomaticTracking

- pds.h, [1392](#)
- pack\_pds\_GetXTRAAutomaticDownload
  - pds.h, [1392](#)
- pack\_pds\_GetXTRANetwork
  - pds.h, [1392](#)
- pack\_pds\_GetXTRAValidity
  - pds.h, [1393](#)
- pack\_pds\_PDSInjectTimeReference
  - pds.h, [1393](#)
- pack\_pds\_PDSInjectTimeReference\_t, [473](#)
  - systemDiscontinuities, [474](#)
  - systemTime, [474](#)
- pack\_pds\_ResetPDSData
  - pds.h, [1393](#)
- pack\_pds\_ResetPDSData\_t, [474](#)
  - pCellDataMask, [474](#)
  - pGPSDataMask, [475](#)
- pack\_pds\_SLQSGetAGPSConfig
  - pds.h, [1396](#)
- pack\_pds\_SLQSGetAGPSConfig\_t, [478](#)
  - pNetworkMode, [479](#)
- pack\_pds\_SLQSGetGPSSStateInfo
  - pds.h, [1397](#)
- pack\_pds\_SLQSPDSDeterminePosition
  - pds.h, [1397](#)
- pack\_pds\_SLQSPDSInjectAbsoluteTimeReference
  - pds.h, [1397](#)
- pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, [479](#)
  - forceFlag, [479](#)
  - timeBase, [479](#)
  - timeMsec, [479](#)
  - timeUncMsec, [480](#)
- pack\_pds\_SLQSPDSInjectPositionData
  - pds.h, [1398](#)
- pack\_pds\_SLQSPDSInjectPositionData\_t, [480](#)
  - pAltitudeWrtEllipsoid, [481](#)
  - pAltitudeWrtSealevel, [481](#)
  - pHorizontalConfidence, [481](#)
  - pHorizontalUncCircular, [481](#)
  - pLatitude, [481](#)
  - pLongitude, [481](#)
  - pPositionSource, [481](#)
  - pTimeStamp, [481](#)
  - pTimeType, [481](#)
  - pVerticalConfidence, [481](#)
  - pVerticalUnc, [481](#)
- pack\_pds\_SLQSSetAGPSConfig
  - pds.h, [1398](#)
- pack\_pds\_SLQSSetAGPSConfig\_t, [482](#)
  - pNetworkMode, [482](#)
  - pServerAddress, [482](#)
  - pServerPort, [482](#)
  - pServerURL, [482](#)
- pack\_pds\_SLQSSetPositionMethodState
  - pds.h, [1399](#)
- pack\_pds\_SLQSSetPositionMethodState\_t, [482](#)
  - pWifiState, [483](#)
- pXtraDataState, [483](#)
- pXtraTimeState, [483](#)
- pack\_pds\_SetEventReportCallback
  - pds.h, [1394](#)
- pack\_pds\_SetEventReportCallback\_t, [475](#)
  - posDataNmea, [475](#)
  - rptPosData, [475](#)
- pack\_pds\_SetPDSDefaults
  - pds.h, [1394](#)
- pack\_pds\_SetPDSDefaults\_t, [475](#)
  - accuracy, [476](#)
  - interval, [476](#)
  - operation, [476](#)
  - timeout, [476](#)
- pack\_pds\_SetPDSSState
  - pds.h, [1395](#)
- pack\_pds\_SetPDSSState\_t, [476](#)
  - enable, [476](#)
- pack\_pds\_SetPortAutomaticTracking
  - pds.h, [1395](#)
- pack\_pds\_SetPortAutomaticTracking\_t, [476](#)
  - bAuto, [477](#)
- pack\_pds\_SetServiceAutomaticTracking
  - pds.h, [1395](#)
- pack\_pds\_SetServiceAutomaticTracking\_t, [477](#)
  - bAuto, [477](#)
- pack\_pds\_SetXTRAAutomaticDownload
  - pds.h, [1396](#)
- pack\_pds\_SetXTRAAutomaticDownload\_t, [477](#)
  - bEnabled, [478](#)
  - interval, [478](#)
- pack\_pds\_SetXTRANetwork
  - pds.h, [1396](#)
- pack\_pds\_SetXTRANetwork\_t, [478](#)
  - preference, [478](#)
- pack\_pds\_StartPDSTrackingSessionExt
  - pds.h, [1399](#)
- pack\_pds\_StartPDSTrackingSessionExt\_t, [483](#)
  - fixAccuracy, [484](#)
  - fixCount, [484](#)
  - fixInterval, [484](#)
  - fixTimeout, [484](#)
  - sessionControl, [484](#)
  - sessionOperation, [484](#)
  - sessionServerOption, [484](#)
  - sessionType, [484](#)
- pack\_pds\_StopPDSTrackingSession
  - pds.h, [1399](#)
- pack\_qmi\_t, [484](#)
  - msgid, [485](#)
  - svc, [485](#)
  - timeout, [485](#)
  - xid, [485](#)
- pack\_qos\_BindDataPort
  - qos.h, [1458](#)
- pack\_qos\_BindDataPort\_t, [485](#)
  - pMuxID, [486](#)
  - pPeripheralEndPointID, [486](#)



- pSIODDataPort, [486](#)
- pack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, [1458](#)
- pack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, [1459](#)
- pack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [486](#)
  - apnId, [486](#)
- pack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, [1460](#)
- pack\_qos\_SLQSQosSwiReadDataStats\_t, [486](#)
  - apnId, [486](#)
- pack\_qos\_SLQSSetQosEventCallback
  - qos.h, [1460](#)
- pack\_qos\_SLQSSetQosEventCallback\_t, [486](#)
  - enable, [487](#)
- pack\_rms\_GetSMSWake
  - rms.h, [1466](#)
- pack\_rms\_SetSMSWake
  - rms.h, [1467](#)
- pack\_rms\_SetSMSWake\_t, [487](#)
  - enabled, [487](#)
  - wake\_mask, [487](#)
- pack\_sar\_SLQSGetRfSarState
  - sar.h, [1468](#)
- pack\_sar\_SLQSSetRfSarState
  - sar.h, [1468](#)
- pack\_sar\_SLQSSetRfSarState\_t, [487](#)
  - RfSarState, [488](#)
- pack\_sms\_GetSMSCAddress
  - sms.h, [1474](#)
- pack\_sms\_SLQSDeleteSMS
  - sms.h, [1476](#)
- pack\_sms\_SLQSDeleteSMS\_t, [490](#)
  - pMessageIndex, [491](#)
  - pMessageMode, [491](#)
  - pMessageTag, [491](#)
  - storageType, [491](#)
- pack\_sms\_SLQSGetIndicationRegister
  - sms.h, [1476](#)
- pack\_sms\_SLQSGetMessageWaiting
  - sms.h, [1476](#)
- pack\_sms\_SLQSGetSMS
  - sms.h, [1477](#)
- pack\_sms\_SLQSGetSMS\_t, [491](#)
  - messageIndex, [492](#)
  - pMessageMode, [492](#)
  - storageType, [492](#)
- pack\_sms\_SLQSGetSMSList
  - sms.h, [1477](#)
- pack\_sms\_SLQSGetSMSList\_t, [492](#)
  - pMessageMode, [493](#)
  - pRequestedTag, [493](#)
  - storageType, [493](#)
- pack\_sms\_SLQSGetSmsBroadcastConfig
  - sms.h, [1477](#)
- pack\_sms\_SLQSGetSmsBroadcastConfig\_t, [492](#)
  - mode, [492](#)
- pack\_sms\_SLQSGetTransLayerInfo
  - sms.h, [1478](#)
- pack\_sms\_SLQSGetTransNWRegInfo
  - sms.h, [1478](#)
- pack\_sms\_SLQSM ModifySMSStatus
  - sms.h, [1478](#)
- pack\_sms\_SLQSM ModifySMSStatus\_t, [493](#)
  - messageIndex, [494](#)
  - messageTag, [494](#)
  - pMessageMode, [494](#)
  - storageType, [494](#)
- pack\_sms\_SLQSSendAsyncSMS
  - sms.h, [1479](#)
- pack\_sms\_SLQSSendAsyncSMS\_t, [494](#)
  - pSendSmsParams, [494](#)
- pack\_sms\_SLQSSetIndicationRegister
  - sms.h, [1479](#)
- pack\_sms\_SLQSSetIndicationRegister\_t, [494](#)
  - pSetIndicationRegReq, [495](#)
- pack\_sms\_SLQSSetSmsBroadcastActivation
  - sms.h, [1479](#)
- pack\_sms\_SLQSSetSmsBroadcastActivation\_t, [495](#)
  - broadcastActivate, [495](#)
  - mode, [495](#)
- pack\_sms\_SLQSSetSmsBroadcastConfig
  - sms.h, [1480](#)
- pack\_sms\_SLQSSetSmsBroadcastConfig\_t, [495](#)
  - mode, [496](#)
  - pBroadcastConfig, [496](#)
  - pCDMABroadcastConfig, [496](#)
- pack\_sms\_SLQSSetSmsStorage
  - sms.h, [1480](#)
- pack\_sms\_SLQSSetSmsStorage\_t, [496](#)
  - smsStorage, [496](#)
- pack\_sms\_SLQSSmsGetMaxStorageSize
  - sms.h, [1481](#)
- pack\_sms\_SLQSSmsGetMaxStorageSize\_t, [497](#)
  - pMaxStorageSizeReq, [497](#)
- pack\_sms\_SLQSSmsGetMessageProtocol
  - sms.h, [1481](#)
- pack\_sms\_SLQSSmsSetRoutes
  - sms.h, [1481](#)
- pack\_sms\_SLQSSmsSetRoutes\_t, [497](#)
  - pSetRoutesReq, [497](#)
- pack\_sms\_SLQSSwiGetSMSStorage
  - sms.h, [1482](#)
- pack\_sms\_SaveSMS
  - sms.h, [1474](#)
- pack\_sms\_SaveSMS\_t, [488](#)
  - messageFormat, [489](#)
  - messageSize, [489](#)
  - pMessage, [489](#)
  - storageType, [489](#)
- pack\_sms\_SendSMS
  - sms.h, [1475](#)
- pack\_sms\_SendSMS\_t, [489](#)
  - messageFormat, [489](#)
  - messageSize, [489](#)
  - pLinktimer, [489](#)

- pMessage, [490](#)
- pack\_sms\_SetNewSMSCallback
  - sms.h, [1475](#)
- pack\_sms\_SetNewSMSCallback\_t, [490](#)
  - status, [490](#)
- pack\_sms\_SetSMSCAddress
  - sms.h, [1475](#)
- pack\_sms\_SetSMSCAddress\_t, [490](#)
  - pSMSCAddress, [490](#)
  - pSMSCType, [490](#)
- pack\_swiaudio\_SLQSGetM2MAVMute
  - swiaudio.h, [1495](#)
- pack\_swiaudio\_SLQSGetM2MAVMute\_t, [498](#)
  - Profile, [499](#)
- pack\_swiaudio\_SLQSGetM2MAudioProfile
  - swiaudio.h, [1494](#)
- pack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [497](#)
  - pGenerator, [498](#)
- pack\_swiaudio\_SLQSGetM2MAudioVolume
  - swiaudio.h, [1494](#)
- pack\_swiaudio\_SLQSGetM2MAudioVolume\_t, [498](#)
  - Generator, [498](#)
  - Profile, [498](#)
- pack\_swiaudio\_SLQSGetM2MSpkrGain
  - swiaudio.h, [1495](#)
- pack\_swiaudio\_SLQSGetM2MSpkrGain\_t, [499](#)
  - Profile, [499](#)
- pack\_swiaudio\_SLQSSetM2MAVMute
  - swiaudio.h, [1497](#)
- pack\_swiaudio\_SLQSSetM2MAVMute\_t, [502](#)
  - EarMute, [503](#)
  - MicMute, [503](#)
  - pCwtMute, [503](#)
  - Profile, [503](#)
- pack\_swiaudio\_SLQSSetM2MAudioAVCFG
  - swiaudio.h, [1496](#)
- pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, [499](#)
  - Device, [500](#)
  - Profile, [500](#)
- pack\_swiaudio\_SLQSSetM2MAudioLPBK
  - swiaudio.h, [1496](#)
- pack\_swiaudio\_SLQSSetM2MAudioLPBK\_t, [500](#)
  - Enable, [500](#)
- pack\_swiaudio\_SLQSSetM2MAudioNVDef
  - swiaudio.h, [1496](#)
- pack\_swiaudio\_SLQSSetM2MAudioProfile
  - swiaudio.h, [1497](#)
- pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, [500](#)
  - pCwtMute, [501](#)
  - pEarMute, [501](#)
  - pGenerator, [501](#)
  - pMicMute, [501](#)
  - pVolume, [501](#)
  - Profile, [501](#)
- pack\_swiaudio\_SLQSSetM2MAudioVolume
  - swiaudio.h, [1497](#)
- pack\_swiaudio\_SLQSSetM2MAudioVolume\_t, [502](#)
  - Generator, [502](#)
- Level, [502](#)
- Profile, [502](#)
- pack\_swiaudio\_SLQSSetM2MSpkrGain
  - swiaudio.h, [1498](#)
- pack\_swiaudio\_SLQSSetM2MSpkrGain\_t, [503](#)
  - Profile, [503](#)
  - Value, [503](#)
- pack\_swiavms\_SLQSAVMSGetSettings
  - swiavms.h, [1504](#)
- pack\_swiavms\_SLQSAVMSGetSettings\_v2
  - swiavms.h, [1505](#)
- pack\_swiavms\_SLQSAVMSSendSelection
  - swiavms.h, [1506](#)
- pack\_swiavms\_SLQSAVMSSendSelection\_t, [504](#)
  - pDeferTime, [504](#)
  - pPackageID, [504](#)
  - pRejectReason, [504](#)
  - selection, [504](#)
- pack\_swiavms\_SLQSAVMSSessionGetInfo
  - swiavms.h, [1506](#)
- pack\_swiavms\_SLQSAVMSSetSettings
  - swiavms.h, [1507](#)
- pack\_swiavms\_SLQSAVMSSetSettings\_t, [504](#)
  - AutoConnect, [506](#)
  - AutoReboot, [506](#)
  - pAPNInfo, [506](#)
  - pConnectionRetryTimers, [506](#)
  - pNotifStore, [506](#)
  - pPeriodInfo, [506](#)
  - pPollingTimer, [506](#)
  - PromptFwDownload, [506](#)
  - PromptFwUpdate, [506](#)
- pack\_swiavms\_SLQSAVMSSetSettings\_v2
  - swiavms.h, [1508](#)
- pack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, [506](#)
  - AutoConnect, [507](#)
  - pAutoReboot, [508](#)
  - pNotifStore, [508](#)
  - pPeriodInfo, [508](#)
  - pPollingTimer, [508](#)
  - PromptFwDownload, [508](#)
  - PromptFwUpdate, [508](#)
- pack\_swiavms\_SLQSAVMSSetSettingsNoAutoReboot-Field
  - swiavms.h, [1509](#)
- pack\_swiavms\_SLQSAVMSStartSession
  - swiavms.h, [1509](#)
- pack\_swiavms\_SLQSAVMSStartSession\_t, [508](#)
  - sessionType, [508](#)
- pack\_swiavms\_SLQSAVMSStopSession
  - swiavms.h, [1510](#)
- pack\_swiavms\_SLQSAVMSStopSession\_t, [508](#)
  - sessionType, [509](#)
- pack\_swiavms\_SLQSAVmsSetEventReport
  - swiavms.h, [1507](#)
- pack\_swidms\_SLQSSwiDmsGetHWWatchdog
  - swidms.h, [1519](#)
- pack\_swidms\_SLQSSwiDmsGetMTU



- swidms.h, [1519](#)
- pack\_swidms\_SLQSSwiDmsGetSecureInfo
  - swidms.h, [1519](#)
- pack\_swidms\_SLQSSwiDmsGetUsbComp
  - swidms.h, [1520](#)
- pack\_swidms\_SLQSSwiDmsGetUsbNetNum
  - swidms.h, [1520](#)
- pack\_swidms\_SLQSSwiDmsSetHWWatchdog
  - swidms.h, [1520](#)
- pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, [509](#)
  - enable, [509](#)
  - resetDelay, [509](#)
  - timeout, [509](#)
- pack\_swidms\_SLQSSwiDmsSetMTU
  - swidms.h, [1521](#)
- pack\_swidms\_SLQSSwiDmsSetMTU\_t, [509](#)
  - MTUSize, [510](#)
- pack\_swidms\_SLQSSwiDmsSetUsbComp
  - swidms.h, [1521](#)
- pack\_swidms\_SLQSSwiDmsSetUsbComp\_t, [510](#)
  - CfgValue, [511](#)
- pack\_swidms\_SLQSSwiDmsSetUsbNetNum
  - swidms.h, [1521](#)
- pack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t, [511](#)
  - nUsbNetNum, [511](#)
- pack\_swiloc\_SwiLocGetAutoStart
  - swiloc.h, [1526](#)
- pack\_swiloc\_SwiLocSetAutoStart
  - swiloc.h, [1526](#)
- pack\_swiloc\_SwiLocSetAutoStart\_t, [512](#)
  - fix\_rate, [513](#)
  - fix\_type, [513](#)
  - function, [513](#)
  - max\_dist, [513](#)
  - max\_time, [513](#)
  - set\_fix\_rate, [513](#)
  - set\_fix\_type, [513](#)
  - set\_function, [513](#)
  - set\_max\_dist, [513](#)
  - set\_max\_time, [513](#)
- pack\_swioama\_SLQSOMADMAAlertCallback
  - swioama.h, [1528](#)
- pack\_swioama\_SLQSOMADMCancelSession
  - swioama.h, [1529](#)
- pack\_swioama\_SLQSOMADMCancelSession\_t, [513](#)
  - sessionType, [514](#)
- pack\_swioama\_SLQSOMADMCancelSessionExt
  - swioamaext.h, [1538](#)
- pack\_swioama\_SLQSOMADMCancelSessionExt\_t, [514](#)
  - sessionType, [514](#)
- pack\_swioama\_SLQSOMADMGetSessionInfo
  - swioama.h, [1530](#)
- pack\_swioama\_SLQSOMADMGetSessionInfo\_t, [514](#)
  - SessionType, [515](#)
- pack\_swioama\_SLQSOMADMGetSessionInfoExt
  - swioamaext.h, [1538](#)
- pack\_swioama\_SLQSOMADMGetSettings
  - swioama.h, [1530](#)
- pack\_swioama\_SLQSOMADMSendSelection
  - swioama.h, [1531](#)
- pack\_swioama\_SLQSOMADMSendSelection\_t, [515](#)
  - pDeferTime, [515](#)
  - pRejectReason, [515](#)
  - selection, [515](#)
- pack\_swioama\_SLQSOMADMSendSelectionExt
  - swioamaext.h, [1539](#)
- pack\_swioama\_SLQSOMADMSendSelectionExt\_t, [515](#)
  - selection, [516](#)
- pack\_swioama\_SLQSOMADMSetSettings
  - swioama.h, [1531](#)
- pack\_swioama\_SLQSOMADMSetSettings\_t, [516](#)
  - FOTAdownload, [517](#)
  - pAutosdm, [517](#)
  - pFwAutoCheck, [517](#)
- pack\_swioama\_SLQSOMADMSetSettingsExt
  - swioamaext.h, [1539](#)
- pack\_swioama\_SLQSOMADMSetSettingsExt\_t, [517](#)
- pack\_swioama\_SLQSOMADMStartSession
  - swioama.h, [1532](#)
- pack\_swioama\_SLQSOMADMStartSession\_t, [518](#)
  - sessionType, [518](#)
- pack\_swioama\_SLQSOMADMStartSessionExt
  - swioamaext.h, [1540](#)
- pack\_swioama\_SLQSOMADMStartSessionExt\_t, [518](#)
  - sessionType, [519](#)
- pack\_tmd\_SLQSTmdDeRegNotMitigationLvl
  - tmd.h, [1546](#)
- pack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, [519](#)
  - mitigationDevID, [519](#)
  - mitigationDevIDLen, [519](#)
- pack\_tmd\_SLQSTmdGetMitigationDevList
  - tmd.h, [1546](#)
- pack\_tmd\_SLQSTmdGetMitigationLvl
  - tmd.h, [1546](#)
- pack\_tmd\_SLQSTmdGetMitigationLvl\_t, [519](#)
  - mitigationDevID, [520](#)
  - mitigationDevIDLen, [520](#)
- pack\_tmd\_SLQSTmdRegNotMitigationLvl
  - tmd.h, [1547](#)
- pack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, [520](#)
  - mitigationDevID, [520](#)
  - mitigationDevIDLen, [520](#)
- pack\_uim\_ChangePin
  - uim.h, [1554](#)
- pack\_uim\_ChangePin\_t, [520](#)
  - changePIN, [521](#)
  - EncryptedPIN1, [521](#)
  - pIndicationToken, [521](#)
  - pKeyReferenceID, [521](#)
  - sessionInfo, [521](#)
  - Tlvresult, [521](#)
- pack\_uim\_GetCardStatus
  - uim.h, [1554](#)
- pack\_uim\_ReadTransparent
  - uim.h, [1554](#)
- pack\_uim\_ReadTransparent\_t, [521](#)

- fileIndex, [522](#)
- pEncryptData, [522](#)
- pIndicationToken, [522](#)
- readTransparent, [522](#)
- sessionInfo, [522](#)
- Tlvresult, [522](#)
- pack\_uim\_SLQSUIMAuthenticate
  - uim.h, [1555](#)
- pack\_uim\_SLQSUIMAuthenticate\_t, [524](#)
  - authData, [524](#)
  - pIndicationToken, [524](#)
  - sessionInfo, [524](#)
- pack\_uim\_SLQSUIMDepersonalization
  - uim.h, [1556](#)
- pack\_uim\_SLQSUIMDepersonalization\_t, [524](#)
  - depersonalisationInfo, [525](#)
- pack\_uim\_SLQSUIMEventRegister
  - uim.h, [1556](#)
- pack\_uim\_SLQSUIMEventRegister\_t, [525](#)
  - eventMask, [525](#)
- pack\_uim\_SLQSUIMGetConfiguration
  - uim.h, [1556](#)
- pack\_uim\_SLQSUIMGetConfiguration\_t, [525](#)
  - pConfigurationMask, [525](#)
- pack\_uim\_SLQSUIMGetFileAttributes
  - uim.h, [1557](#)
- pack\_uim\_SLQSUIMGetFileAttributes\_t, [526](#)
  - fileIndex, [526](#)
  - pIndicationToken, [526](#)
  - sessionInfo, [526](#)
- pack\_uim\_SLQSUIMGetServiceStatus
  - uim.h, [1557](#)
- pack\_uim\_SLQSUIMGetServiceStatus\_t, [526](#)
  - capMask, [527](#)
  - sessionInfo, [527](#)
- pack\_uim\_SLQSUIMGetSlotsStatus
  - uim.h, [1557](#)
- pack\_uim\_SLQSUIMPowerDown
  - uim.h, [1558](#)
- pack\_uim\_SLQSUIMPowerDown\_t, [527](#)
  - slot, [527](#)
- pack\_uim\_SLQSUIMPowerUp
  - uim.h, [1558](#)
- pack\_uim\_SLQSUIMPowerUp\_t, [527](#)
  - plgnoreHotSwapSwitch, [528](#)
  - slot, [528](#)
- pack\_uim\_SLQSUIMReadRecord
  - uim.h, [1559](#)
- pack\_uim\_SLQSUIMReadRecord\_t, [528](#)
  - fileIndex, [529](#)
  - pIndicationToken, [529](#)
  - pLastRecord, [529](#)
  - readRecord, [529](#)
  - sessionInfo, [529](#)
- pack\_uim\_SLQSUIMRefreshComplete
  - uim.h, [1559](#)
- pack\_uim\_SLQSUIMRefreshComplete\_t, [529](#)
  - refreshComplete, [529](#)
  - sessionInfo, [529](#)
- pack\_uim\_SLQSUIMRefreshGetLastEvent
  - uim.h, [1559](#)
- pack\_uim\_SLQSUIMRefreshGetLastEvent\_t, [529](#)
  - sessionInfo, [530](#)
- pack\_uim\_SLQSUIMRefreshOK
  - uim.h, [1560](#)
- pack\_uim\_SLQSUIMRefreshOK\_t, [530](#)
  - OKtoRefresh, [530](#)
  - sessionInfo, [530](#)
- pack\_uim\_SLQSUIMRefreshRegister
  - uim.h, [1560](#)
- pack\_uim\_SLQSUIMRefreshRegister\_t, [530](#)
  - regRefresh, [531](#)
  - sessionInfo, [531](#)
- pack\_uim\_SLQSUIMReset
  - uim.h, [1560](#)
- pack\_uim\_SLQSUIMSetServiceStatus
  - uim.h, [1561](#)
- pack\_uim\_SLQSUIMSetServiceStatus\_t, [531](#)
  - pFDNStatus, [531](#)
  - sessionInfo, [531](#)
- pack\_uim\_SLQSUIMSwitchSlot
  - uim.h, [1561](#)
- pack\_uim\_SLQSUIMSwitchSlot\_t, [531](#)
  - bLogicalSlot, [532](#)
  - ulPhysicalSlot, [532](#)
- pack\_uim\_SLQSUIMWriteRecord
  - uim.h, [1561](#)
- pack\_uim\_SLQSUIMWriteRecord\_t, [532](#)
  - fileIndex, [533](#)
  - pIndicationToken, [533](#)
  - sessionInfo, [533](#)
  - writeRecord, [533](#)
- pack\_uim\_SLQSUIMWriteTransparent
  - uim.h, [1562](#)
- pack\_uim\_SLQSUIMWriteTransparent\_t, [533](#)
  - fileIndex, [534](#)
  - pIndicationToken, [534](#)
  - sessionInfo, [534](#)
  - writeTransparent, [534](#)
- pack\_uim\_SetPinProtection
  - uim.h, [1555](#)
- pack\_uim\_SetPinProtection\_t, [523](#)
  - EncryptedPIN1, [523](#)
  - pIndicationToken, [523](#)
  - pKeyReferenceID, [523](#)
  - pinProtection, [523](#)
  - sessionInfo, [523](#)
  - Tlvresult, [523](#)
- pack\_uim\_UnblockPin
  - uim.h, [1562](#)
- pack\_uim\_UnblockPin\_t, [534](#)
  - EncryptedPIN1, [534](#)
  - pIndicationToken, [535](#)
  - pKeyReferenceID, [535](#)
  - pinProtection, [535](#)
  - sessionInfo, [535](#)

- Tlvresult, [535](#)
- pack\_uim\_VerifyPin
  - uim.h, [1563](#)
- pack\_uim\_VerifyPin\_t, [535](#)
  - pEncryptedPIN1, [536](#)
  - pIndicationToken, [536](#)
  - pKeyReferenceID, [536](#)
  - sessionInfo, [536](#)
  - Tlvresult, [536](#)
  - verifyPIN, [536](#)
- pack\_voice\_AnswerUSSD
  - voice.h, [1580](#)
- pack\_voice\_AnswerUSSD\_t, [536](#)
  - pInfo, [536](#)
- pack\_voice\_CancelUSSD
  - voice.h, [1581](#)
- pack\_voice\_OriginateUSSD
  - voice.h, [1581](#)
- pack\_voice\_OriginateUSSD\_t, [536](#)
  - pInfo, [537](#)
- pack\_voice\_SLQSOriinateUSSD
  - voice.h, [1581](#)
- pack\_voice\_SLQSOriinateUSSD\_t, [537](#)
  - ussDCS, [537](#)
  - ussData, [537](#)
  - ussLen, [537](#)
- pack\_voice\_SLQSVoiceALSSelectLine
  - voice.h, [1582](#)
- pack\_voice\_SLQSVoiceALSSelectLine\_t, [537](#)
  - lineValue, [538](#)
- pack\_voice\_SLQSVoiceALSSetLineSwitching
  - voice.h, [1582](#)
- pack\_voice\_SLQSVoiceALSSetLineSwitching\_t, [538](#)
  - switchOption, [538](#)
- pack\_voice\_SLQSVoiceAnswerCall
  - voice.h, [1582](#)
- pack\_voice\_SLQSVoiceAnswerCall\_t, [538](#)
  - pCallId, [539](#)
- pack\_voice\_SLQSVoiceBindSubscription
  - voice.h, [1583](#)
- pack\_voice\_SLQSVoiceBindSubscription\_t, [539](#)
  - subsType, [539](#)
- pack\_voice\_SLQSVoiceBurstDTMF
  - voice.h, [1583](#)
- pack\_voice\_SLQSVoiceBurstDTMF\_t, [539](#)
- pack\_voice\_SLQSVoiceDialCall
  - voice.h, [1584](#)
- pack\_voice\_SLQSVoiceDialCall\_t, [540](#)
  - callNumber, [541](#)
  - pCLIRType, [541](#)
  - pCUGInfo, [541](#)
  - pCallPartySubAdd, [541](#)
  - pCallType, [541](#)
  - pEmergencyCategory, [541](#)
  - pSvcType, [541](#)
  - pUUSInfo, [541](#)
- pack\_voice\_SLQSVoiceEndCall
  - voice.h, [1584](#)
- pack\_voice\_SLQSVoiceEndCall\_t, [541](#)
  - pCallId, [542](#)
- pack\_voice\_SLQSVoiceGetAllCallInfo
  - voice.h, [1584](#)
- pack\_voice\_SLQSVoiceGetCLIP
  - voice.h, [1586](#)
- pack\_voice\_SLQSVoiceGetCLIR
  - voice.h, [1587](#)
- pack\_voice\_SLQSVoiceGetCNAP
  - voice.h, [1587](#)
- pack\_voice\_SLQSVoiceGetCOLP
  - voice.h, [1587](#)
- pack\_voice\_SLQSVoiceGetCOLR
  - voice.h, [1588](#)
- pack\_voice\_SLQSVoiceGetCallBarring
  - voice.h, [1585](#)
- pack\_voice\_SLQSVoiceGetCallBarring\_t, [542](#)
  - pSvcClass, [542](#)
  - reason, [542](#)
- pack\_voice\_SLQSVoiceGetCallForwardingStatus
  - voice.h, [1585](#)
- pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t, [543](#)
  - pSvcClass, [543](#)
  - Reason, [543](#)
- pack\_voice\_SLQSVoiceGetCallInfo
  - voice.h, [1585](#)
- pack\_voice\_SLQSVoiceGetCallInfo\_t, [543](#)
  - callID, [544](#)
- pack\_voice\_SLQSVoiceGetCallWaiting
  - voice.h, [1586](#)
- pack\_voice\_SLQSVoiceGetCallWaiting\_t, [544](#)
  - pSvcClass, [544](#)
- pack\_voice\_SLQSVoiceGetConfig
  - voice.h, [1588](#)
- pack\_voice\_SLQSVoiceGetConfig\_t, [544](#)
  - pAMRStatus, [545](#)
  - pAirTimer, [545](#)
  - pAutoAnswer, [546](#)
  - pNamID, [546](#)
  - pPrefVoicePrivacy, [546](#)
  - pPrefVoiceSO, [546](#)
  - pRoamTimer, [546](#)
  - pTTYMode, [546](#)
  - pVoiceDomainPref, [546](#)
- pack\_voice\_SLQSVoiceIndicationRegister
  - voice.h, [1588](#)
- pack\_voice\_SLQSVoiceIndicationRegister\_t, [546](#)
  - pRegDTMFEvents, [547](#)
  - pRegVoicePrivacyEvents, [547](#)
  - pSuppsNotifEvents, [547](#)
- pack\_voice\_SLQSVoiceManageCalls
  - voice.h, [1589](#)
- pack\_voice\_SLQSVoiceManageCalls\_t, [547](#)
  - pCallID, [548](#)
  - SUPSType, [548](#)
- pack\_voice\_SLQSVoiceOrigUSSDNoWait
  - voice.h, [1589](#)
- pack\_voice\_SLQSVoiceOrigUSSDNoWait\_t, [548](#)

- pack\_voice\_SLQSVoiceSendFlash
  - voice.h, [1589](#)
- pack\_voice\_SLQSVoiceSendFlash\_t, [548](#)
  - pCallID, [549](#)
  - pFlashPayLd, [549](#)
  - pFlashType, [549](#)
- pack\_voice\_SLQSVoiceSetCallBarringPassword
  - voice.h, [1590](#)
- pack\_voice\_SLQSVoiceSetCallBarringPassword\_t, [549](#)
  - newPasswd, [550](#)
  - newPasswdAgain, [550](#)
  - oldPasswd, [550](#)
  - Reason, [550](#)
- pack\_voice\_SLQSVoiceSetConfig
  - voice.h, [1590](#)
- pack\_voice\_SLQSVoiceSetConfig\_t, [550](#)
  - pAirTimerConfig, [551](#)
  - pAutoAnswer, [551](#)
  - pPrefVoiceDomain, [551](#)
  - pPrefVoiceSO, [551](#)
  - pRoamTimerConfig, [551](#)
  - pTTYMode, [551](#)
- pack\_voice\_SLQSVoiceSetPreferredPrivacy
  - voice.h, [1590](#)
- pack\_voice\_SLQSVoiceSetPreferredPrivacy\_t, [551](#)
  - privacyPref, [551](#)
- pack\_voice\_SLQSVoiceSetSUPSService
  - voice.h, [1591](#)
- pack\_voice\_SLQSVoiceSetSUPSService\_t, [551](#)
  - pCallBarringPasswd, [554](#)
  - pCallForwardingNumber, [554](#)
  - pServiceClass, [554](#)
  - pTimerVal, [554](#)
  - reason, [554](#)
  - voiceSvc, [554](#)
- pack\_voice\_SLQSVoiceStartContDTMF
  - voice.h, [1591](#)
- pack\_voice\_SLQSVoiceStartContDTMF\_t, [554](#)
  - DTMFdigit, [554](#)
  - pCallID, [554](#)
- pack\_voice\_SLQSVoiceStopContDTMF
  - voice.h, [1592](#)
- pack\_voice\_SLQSVoiceStopContDTMF\_t, [555](#)
  - callID, [555](#)
- pack\_wds\_DHCPv4ClientLeaseChange
  - wds.h, [1616](#)
- pack\_wds\_DHCPv4ClientLeaseChange\_t, [555](#)
  - pEnableNotification, [555](#)
- pack\_wds\_GetAutoconnect
  - wds.h, [1617](#)
- pack\_wds\_GetByteTotals
  - wds.h, [1617](#)
- pack\_wds\_GetConnectionRate
  - wds.h, [1617](#)
- pack\_wds\_GetDataBearerTechnology
  - wds.h, [1618](#)
- pack\_wds\_GetDefaultProfile
  - wds.h, [1618](#)
- pack\_wds\_GetDefaultProfile\_t, [555](#)
  - profiletype, [556](#)
- pack\_wds\_GetDefaultProfileNum
  - wds.h, [1619](#)
- pack\_wds\_GetDefaultProfileNum\_t, [556](#)
  - family, [556](#)
  - type, [556](#)
- pack\_wds\_GetDefaultProfileV2
  - wds.h, [1619](#)
- pack\_wds\_GetDefaultProfileV2\_t, [556](#)
  - profiletype, [557](#)
- pack\_wds\_GetDormancyState
  - wds.h, [1619](#)
- pack\_wds\_GetDormancyState\_t, [557](#)
- pack\_wds\_GetLastMobileIPError
  - wds.h, [1620](#)
- pack\_wds\_GetLastMobileIPError\_t, [557](#)
- pack\_wds\_GetMobileIP
  - wds.h, [1620](#)
- pack\_wds\_GetMobileIP\_t, [557](#)
- pack\_wds\_GetMobileIPProfile
  - wds.h, [1621](#)
- pack\_wds\_GetMobileIPProfile\_t, [558](#)
  - index, [558](#)
- pack\_wds\_GetPacketStatistics
  - wds.h, [1621](#)
- pack\_wds\_GetPacketStatistics\_t, [558](#)
  - pStatMask, [558](#)
- pack\_wds\_GetPacketStatus
  - wds.h, [1622](#)
- pack\_wds\_GetPacketStatus\_t, [558](#)
  - statmask, [559](#)
- pack\_wds\_GetSessionDuration
  - wds.h, [1622](#)
- pack\_wds\_GetSessionDuration\_t, [559](#)
- pack\_wds\_GetSessionDurationV2
  - wds.h, [1622](#)
- pack\_wds\_GetSessionState
  - wds.h, [1623](#)
- pack\_wds\_RMSetTransferStatistics
  - wds.h, [1623](#)
- pack\_wds\_RMSetTransferStatistics\_t, [559](#)
  - RmTrasnferStaticsReq, [560](#)
- pack\_wds\_SLQSCreateProfile
  - wds.h, [1627](#)
- pack\_wds\_SLQSCreateProfile\_t, [566](#)
  - pCurProfile, [567](#)
  - pProfileId, [567](#)
  - pProfileType, [567](#)
- pack\_wds\_SLQSDeleteProfile
  - wds.h, [1627](#)
- pack\_wds\_SLQSDeleteProfile\_t, [567](#)
  - profileIndex, [567](#)
  - profileType, [568](#)
- pack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, [1627](#)
- pack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, [1628](#)

- pack\_wds\_SLQSGetCurrDataSystemStat\_t, 568
- pack\_wds\_SLQSGetCurrentChannelRate
  - wds.h, 1628
- pack\_wds\_SLQSGetDUNCallInfo
  - wds.h, 1629
- pack\_wds\_SLQSGetDUNCallInfo\_t, 568
  - Mask, 569
  - pReportChannelRate, 569
  - pReportConnStatus, 569
  - pReportDataBearerTech, 569
  - pReportDormStatus, 569
  - pTransferStatInd, 569
- pack\_wds\_SLQSGetDataBearerTechnology
  - wds.h, 1629
- pack\_wds\_SLQSGetDataBearerTechnology\_t, 568
- pack\_wds\_SLQSGetProfileSettings
  - wds.h, 1630
- pack\_wds\_SLQSGetProfileSettings\_t, 570
  - ProfileId, 570
  - ProfileType, 570
- pack\_wds\_SLQSGetProfileSettingsV2
  - wds.h, 1630
- pack\_wds\_SLQSGetRuntimeSettings
  - wds.h, 1630
- pack\_wds\_SLQSGetRuntimeSettings\_t, 570
  - pReqSettings, 571
- pack\_wds\_SLQSModifyProfile
  - wds.h, 1631
- pack\_wds\_SLQSModifyProfile\_t, 571
  - curProfile, 572
  - pProfileId, 572
  - pProfileType, 572
- pack\_wds\_SLQSResetPacketStatics
  - wds.h, 1631
- pack\_wds\_SLQSSGetDHCPv4ClientConfig
  - wds.h, 1633
- pack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, 575
  - pProfileId, 575
- pack\_wds\_SLQSSGetLoopback
  - wds.h, 1633
- pack\_wds\_SLQSSSetDHCPv4ClientConfig
  - wds.h, 1634
- pack\_wds\_SLQSSSetDHCPv4ClientConfig\_t, 575
  - pHwConfig, 576
  - pProfileId, 576
- pack\_wds\_SLQSSSetLoopback
  - wds.h, 1634
- pack\_wds\_SLQSSSetLoopback\_t, 576
  - loopbackMode, 576
  - loopbackMultiplier, 576
- pack\_wds\_SLQSSet3GPPConfigItem
  - wds.h, 1632
- pack\_wds\_SLQSSet3GPPConfigItem\_t, 572
  - p3gppRelease, 573
  - pProfileList, 573
- pack\_wds\_SLQSSetIPFamilyPreference
  - wds.h, 1632
- pack\_wds\_SLQSSetIPFamilyPreference\_t, 574
  - IPFamilyPreference, 574
- pack\_wds\_SLQSSetWdsEventCallback
  - wds.h, 1632
- pack\_wds\_SLQSSetWdsEventCallback\_t, 574
  - currentDataBearer, 575
  - dataBearer, 575
  - dataBearerTechExt, 575
  - dataSystemStatus, 575
  - dormancyStatus, 575
  - interval, 575
  - mobileIP, 575
  - transferStats, 575
- pack\_wds\_SLQSStartDataSession
  - wds.h, 1634
- pack\_wds\_SLQSStartDataSession\_t, 576
  - pAuth, 577
  - pPass, 577
  - pTech, 577
  - pUser, 577
  - pprofileid3gpp, 577
  - pprofileid3gpp2, 577
- pack\_wds\_SLQSStopDataSession
  - wds.h, 1635
- pack\_wds\_SLQSStopDataSession\_t, 578
  - psid, 578
- pack\_wds\_SLQSSwiProfileChangeCallback
  - wds.h, 1635
- pack\_wds\_SLQSSwiProfileChangeCallback\_t, 578
  - pProfileChangeInd, 578
- pack\_wds\_SLQSWdsGoActive
  - wds.h, 1636
- pack\_wds\_SLQSWdsGoDormant
  - wds.h, 1636
- pack\_wds\_SLQSWdsSetEventReport
  - wds.h, 1637
- pack\_wds\_SLQSWdsSetEventReport\_t, 578
  - pCurrChannelRateInd, 580
  - pCurrDataBearerTechInd, 580
  - pCurrPrefDataSysInd, 580
  - pDataBearerTechInd, 580
  - pDataCallStatusChangeInd, 580
  - pDataSystemStatusChangeInd, 580
  - pDormancyStatusInd, 580
  - pMIPStatusInd, 580
  - pTransferStatInd, 580
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings
  - wds.h, 1637
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 580
  - contextId, 581
  - contextType, 581
- pack\_wds\_SetAutoconnect
  - wds.h, 1624
- pack\_wds\_SetAutoconnect\_t, 560
  - acroamsetting, 560
  - acsetting, 560
- pack\_wds\_SetDefaultProfile
  - wds.h, 1624
- pack\_wds\_SetDefaultProfile\_t, 560

- authentication, 562
- ipAddress, 562
- pApnname, 562
- pName, 562
- pPassword, 562
- pUsername, 562
- pdpType, 562
- primaryDNS, 562
- profileType, 562
- secondaryDNS, 562
- pack\_wds\_SetDefaultProfileNum
  - wds.h, 1624
- pack\_wds\_SetDefaultProfileNum\_t, 562
  - family, 563
  - index, 563
  - type, 563
- pack\_wds\_SetMobileIP
  - wds.h, 1625
- pack\_wds\_SetMobileIP\_t, 563
  - mode, 563
- pack\_wds\_SetMobileIPParameters
  - wds.h, 1625
- pack\_wds\_SetMobileIPParameters\_t, 563
  - pHA2002bis, 564
  - pHAAuthenticator, 564
  - pMode, 564
  - pReRegPeriod, 564
  - pReRegTraffic, 564
  - pRetryInterval, 564
  - pRetryLimit, 564
  - pSPC, 564
- pack\_wds\_SetMobileIPProfile
  - wds.h, 1626
- pack\_wds\_SetMobileIPProfile\_t, 564
  - index, 566
  - pAAASPI, 566
  - pAddress, 566
  - pEnabled, 566
  - pHASPI, 566
  - pMNAAA, 566
  - pMNHA, 566
  - pNAI, 566
  - pPrimaryHA, 566
  - pRevTunneling, 566
  - pSecondaryHA, 566
  - spc, 566
- pack\_wds\_SetMuxID
  - wds.h, 1626
- PackCreateProfileOut, 581
  - ExtErrorCode, 581
  - ProfileIndex, 581
  - ProfileType, 581
- PackSwiAVMSSettingsAPNInfo, 584
  - bAPNLength, 585
  - bPWDLength, 585
  - bUnameLength, 585
  - pAPN, 585
  - pPWD, 585
  - pUname, 585
- PackSwiAVMSSettingsConnectionRetryTimers, 585
  - Timers, 586
- PackSwiAVMSSettingsPeriodsInfo, 586
  - max, 586
  - min, 586
- PackSwiAvmsSetSettingsAPNInfo, 582
  - bAPNLength, 583
  - bPWDLength, 583
  - bUnameLength, 583
  - szAPN, 583
  - szPWD, 583
  - szUname, 583
- PackSwiAvmsSetSettingsConnectionRetryTimers, 583
  - Timers, 584
- PackSwiAvmsSetSettingsPeriodInfo, 584
  - ulMax, 584
  - ulMin, 584
- package\_name
  - unpack\_omaDmFotaTlv\_t, 891
- packageIdStr
  - \_litefw\_FirmwareFileInfo, 36
- packageid\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 678
- packetZone
  - nas\_CDMASysInfo, 185
- packetZoneValid
  - nas\_CDMASysInfo, 185
- packgetDyingGaspCfg, 581
  - pDestSMSContent, 582
  - pDestSMSNum, 582
- packgetDyingGaspStatistics, 582
  - pSMSAttemptedFlag, 582
  - pTimeStamp, 582
- ParamPresenceMask
  - pack\_dms\_UIMGetICCID\_t, 395
  - unpack\_audio\_SLQSGetAudioPathConfig\_t, 657
  - unpack\_audio\_SLQSGetAudioProfile\_t, 658
  - unpack\_audio\_SLQSGetAudioVolTLBConfig\_t, 659
  - unpack\_audio\_SLQSSetAudioVolTLBConfig\_t, 659
  - unpack\_cat\_SetCatEventCallback\_ind\_t, 660
  - unpack\_cat\_SetCATEventCallback\_t, 661
  - unpack\_dms\_GetActivationState\_t, 662
  - unpack\_dms\_GetBandCapability\_t, 664
  - unpack\_dms\_GetCrashAction\_t, 664
  - unpack\_dms\_GetCustFeature\_t, 667
  - unpack\_dms\_GetCustFeaturesV2\_t, 668
  - unpack\_dms\_GetDeviceCap\_t, 669
  - unpack\_dms\_GetDeviceCapabilities\_t, 671
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, 673
  - unpack\_dms\_GetDeviceHardwareRev\_t, 674
  - unpack\_dms\_GetDeviceMfr\_t, 675
  - unpack\_dms\_GetDeviceSerialNumbers\_t, 677
  - unpack\_dms\_GetFirmwareInfo\_t, 678
  - unpack\_dms\_GetFirmwareRevision\_t, 679
  - unpack\_dms\_GetFirmwareRevisions\_t, 681



- [unpack\\_dms\\_GetFSN\\_t, 681](#)
- [unpack\\_dms\\_GetHardwareRevision\\_t, 682](#)
- [unpack\\_dms\\_GetIMSI\\_t, 683](#)
- [unpack\\_dms\\_GetManufacturer\\_t, 683](#)
- [unpack\\_dms\\_GetModelID\\_t, 684](#)
- [unpack\\_dms\\_GetNetworkTime\\_t, 685](#)
- [unpack\\_dms\\_GetNetworkTimeV2\\_t, 686](#)
- [unpack\\_dms\\_GetOfflineReason\\_t, 687](#)
- [unpack\\_dms\\_GetPower\\_t, 688](#)
- [unpack\\_dms\\_GetPRLVersion\\_t, 689](#)
- [unpack\\_dms\\_GetSerialNumbers\\_t, 690](#)
- [unpack\\_dms\\_GetUSBComp\\_t, 692](#)
- [unpack\\_dms\\_GetVoiceNumber\\_t, 693](#)
- [unpack\\_dms\\_PSMCfgChange\\_ind\\_t, 694](#)
- [unpack\\_dms\\_ResetToFactoryDefaults\\_t, 695](#)
- [unpack\\_dms\\_SetActivationStatusCallback\\_t, 695](#)
- [unpack\\_dms\\_SetCrashAction\\_t, 696](#)
- [unpack\\_dms\\_SetCustFeature\\_t, 696](#)
- [unpack\\_dms\\_SetCustFeaturesV2\\_t, 697](#)
- [unpack\\_dms\\_SetEventReport\\_ind\\_t, 698](#)
- [unpack\\_dms\\_SetEventReport\\_t, 698](#)
- [unpack\\_dms\\_SetFirmwarePreference\\_t, 699](#)
- [unpack\\_dms\\_SetIndicationRegister\\_t, 699](#)
- [unpack\\_dms\\_SetPower\\_t, 700](#)
- [unpack\\_dms\\_SetUSBComp\\_t, 700](#)
- [unpack\\_dms\\_SLQSDmsSwiGetPCInfo\\_t, 703](#)
- [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_ind\\_t, 704](#)
- [unpack\\_dms\\_SLQSDmsSwiGetResetInfo\\_t, 705](#)
- [unpack\\_dms\\_SLQSDmsSwiGetUimSelection\\_t, 706](#)
- [unpack\\_dms\\_SLQSDmsSwiIndicationRegister\\_t, 707](#)
- [unpack\\_dms\\_SLQSGetBandCapability\\_t, 711](#)
- [unpack\\_dms\\_SLQSGetBandCapabilityExt\\_t, 715](#)
- [unpack\\_dms\\_SLQSGetERIFile\\_t, 716](#)
- [unpack\\_dms\\_SLQSGetPowerSaveModeConfig\\_t, 717](#)
- [unpack\\_dms\\_SLQSSetPowerSaveModeConfig\\_t, 718](#)
- [unpack\\_dms\\_SLQSSwiClearDyingGaspStatistics\\_t, 718](#)
- [unpack\\_dms\\_SLQSSwiGetCrashInfo\\_t, 719](#)
- [unpack\\_dms\\_SLQSSwiGetDyingGaspCfg\\_t, 720](#)
- [unpack\\_dms\\_SLQSSwiGetDyingGaspStatistics\\_t, 720](#)
- [unpack\\_dms\\_SLQSSwiGetFirmwareCurr\\_t, 721](#)
- [unpack\\_dms\\_SLQSSwiGetFwUpdateStatus\\_t, 723](#)
- [unpack\\_dms\\_SLQSSwiGetHostDevInfo\\_t, 724](#)
- [unpack\\_dms\\_SLQSSwiGetOSInfo\\_t, 725](#)
- [unpack\\_dms\\_SLQSSwiGetSerialNoExt\\_t, 726](#)
- [unpack\\_dms\\_SLQSSwiSetDyingGaspCfg\\_t, 726](#)
- [unpack\\_dms\\_SLQSSwiSetHostDevInfo\\_t, 727](#)
- [unpack\\_dms\\_SLQSSwiSetOSInfo\\_t, 727](#)
- [unpack\\_dms\\_SLQSUIMGetState\\_t, 728](#)
- [unpack\\_dms\\_SwiEventReportCallback\\_ind\\_t, 729](#)
- [unpack\\_dms\\_SwiSetEventReport\\_t, 729](#)
- [unpack\\_dms\\_SwiUimSelect\\_t, 730](#)
- [unpack\\_dms\\_UIMGetControlKeyStatus\\_t, 731](#)
- [unpack\\_dms\\_UIMGetICCID\\_t, 731](#)
- [unpack\\_dms\\_UIMGetPINStatus\\_t, 733](#)
- [unpack\\_dms\\_UIMSetControlKeyProtection\\_t, 734](#)
- [unpack\\_dms\\_UIMSetPINProtection\\_t, 735](#)
- [unpack\\_dms\\_UIMUnlockControlKey\\_t, 735](#)
- [unpack\\_fms\\_GetImagesPreference\\_t, 736](#)
- [unpack\\_fms\\_GetStoredImages\\_t, 737](#)
- [unpack\\_fms\\_SetImagesPreference\\_t, 737](#)
- [unpack\\_ims\\_SLQSGetIMSSMSConfig\\_t, 738](#)
- [unpack\\_ims\\_SLQSGetIMSUserConfig\\_t, 739](#)
- [unpack\\_ims\\_SLQSGetIMSVoIPConfig\\_t, 742](#)
- [unpack\\_ims\\_SLQSGetRegMgrConfig\\_t, 743](#)
- [unpack\\_ims\\_SLQSGetSIPConfig\\_t, 744](#)
- [unpack\\_ims\\_SLQSSetRegMgrCfgCallback\\_ind\\_t, 745](#)
- [unpack\\_ims\\_SLQSSetIMSSMSConfig\\_t, 746](#)
- [unpack\\_ims\\_SLQSSetIMSUserConfig\\_t, 746](#)
- [unpack\\_ims\\_SLQSSetIMSVoIPConfig\\_t, 747](#)
- [unpack\\_ims\\_SLQSSetRegMgrConfig\\_t, 747](#)
- [unpack\\_ims\\_SLQSSetSIPConfig\\_t, 748](#)
- [unpack\\_ims\\_SLQSSIPCfgCallback\\_ind\\_t, 749](#)
- [unpack\\_ims\\_SLQSSMSCfgCallback\\_ind\\_t, 750](#)
- [unpack\\_ims\\_SLQSSetUserCfgCallback\\_ind\\_t, 751](#)
- [unpack\\_ims\\_SLQSVoIPCfgCallback\\_ind\\_t, 753](#)
- [unpack\\_imsa\\_SLQSGetIMSARegStatus\\_t, 754](#)
- [unpack\\_imsa\\_SLQSGetIMSAServiceStatus\\_t, 756](#)
- [unpack\\_imsa\\_SLQSImsaPdpStatusCallback\\_ind\\_t, 757](#)
- [unpack\\_imsa\\_SLQSImsaRatStatusCallback\\_ind\\_t, 758](#)
- [unpack\\_imsa\\_SLQSImsaRegStatusCallback\\_ind\\_t, 758](#)
- [unpack\\_imsa\\_SLQSImsaSvcStatusCallback\\_ind\\_t, 760](#)
- [unpack\\_loc\\_BestAvailPos\\_ind\\_t, 766](#)
- [unpack\\_loc\\_CradleMountCallback\\_ind\\_t, 767](#)
- [unpack\\_loc\\_Delete\\_Assist\\_Data\\_t, 768](#)
- [unpack\\_loc\\_DeleteAssistData\\_ind\\_t, 769](#)
- [unpack\\_loc\\_EngineState\\_ind\\_t, 770](#)
- [unpack\\_loc\\_EventNMEA\\_ind\\_t, 770](#)
- [unpack\\_loc\\_EventRegister\\_t, 771](#)
- [unpack\\_loc\\_EventTimeSyncCallback\\_ind\\_t, 771](#)
- [unpack\\_loc\\_FixCriteria\\_ind\\_t, 772](#)
- [unpack\\_loc\\_GetOpMode\\_ind\\_t, 774](#)
- [unpack\\_loc\\_GetServer\\_ind\\_t, 775](#)
- [unpack\\_loc\\_GnssSvInfo\\_ind\\_t, 776](#)
- [unpack\\_loc\\_InjectPositionCallback\\_ind\\_t, 777](#)
- [unpack\\_loc\\_InjectSensorDataCallback\\_ind\\_t, 779](#)
- [unpack\\_loc\\_InjectTimeSyncDataCallback\\_ind\\_t, 779](#)
- [unpack\\_loc\\_InjectUTCTimeCallback\\_ind\\_t, 780](#)
- [unpack\\_loc\\_PositionRpt\\_ind\\_t, 786](#)
- [unpack\\_loc\\_SensorStreamingCallback\\_ind\\_t, 787](#)
- [unpack\\_loc\\_SetExtPowerConfig\\_ind\\_t, 788](#)
- [unpack\\_loc\\_SetExtPowerState\\_t, 789](#)
- [unpack\\_loc\\_SetOperationMode\\_ind\\_t, 790](#)
- [unpack\\_loc\\_SetOperationMode\\_t, 790](#)
- [unpack\\_loc\\_SetServer\\_ind\\_t, 791](#)

- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, 792
- unpack\_loc\_SLQSLOCGetOpMode\_t, 792
- unpack\_loc\_Start\_t, 793
- unpack\_loc\_Stop\_t, 793
- unpack\_nas\_GetACCOLC\_t, 794
- unpack\_nas\_GetANAAAAAuthenticationStatus\_t, 794
- unpack\_nas\_GetCDMANetworkParameters\_t, 796
- unpack\_nas\_GetHomeNetwork3GPP2\_t, 799
- unpack\_nas\_GetHomeNetwork\_t, 800
- unpack\_nas\_GetNetworkPreference\_t, 801
- unpack\_nas\_GetRFInfo\_t, 802
- unpack\_nas\_GetServingNetwork\_t, 805
- unpack\_nas\_GetServingNetworkCapabilities\_t, 806
- unpack\_nas\_GetSignalStrengths\_t, 806
- unpack\_nas\_PerformNetworkScan\_t, 808
- unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 809
- unpack\_nas\_SetEventReportInd\_t, 810
- unpack\_nas\_SetNasLTECphyCalIndCallback\_ind\_t, 811
- unpack\_nas\_SetNetworkPreference\_t, 812
- unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, 812
- unpack\_nas\_SetServingSystemCallback\_ind\_t, 813
- unpack\_nas\_SLQSGetErrorRate\_t, 814
- unpack\_nas\_SLQSGetHomeNetwork\_t, 816
- unpack\_nas\_SLqsGetLTECphyCAInfo\_t, 816
- unpack\_nas\_SLQSGetNetworkTime\_t, 817
- unpack\_nas\_SLQSGetOperatorNameData\_t, 818
- unpack\_nas\_SLQSGetPLMNName\_t, 821
- unpack\_nas\_SLQSGetServingSystem\_t, 826
- unpack\_nas\_SLQSGetServingSystemV2\_t, 831
- unpack\_nas\_SLQSGetSignalStrength\_t, 833
- unpack\_nas\_SLQSGetSysInfo\_t, 837
- unpack\_nas\_SLQSGetSysInfoV2\_t, 841
- unpack\_nas\_SLQSGetSysSelectionPref\_t, 847
- unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, 853
- unpack\_nas\_SLQSNasEdrxChangeInfoCallBack\_ind\_t, 855
- unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, 856
- unpack\_nas\_SLQSNasGetCellLocationInfo\_t, 857
- unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, 859
- unpack\_nas\_SLQSNASGetDRXParams\_t, 860
- unpack\_nas\_SLQSNASGetDRXParamsExt\_t, 862
- unpack\_nas\_SLQSNASGetForbiddenNetworks\_t, 862
- unpack\_nas\_SLQSNasGetHDRColorCode\_t, 863
- unpack\_nas\_SLQSNasGetSigInfo\_t, 865
- unpack\_nas\_SLQSNasGetTxRxInfo\_t, 866
- unpack\_nas\_SLQSNasNetworkRejectCallback\_ind\_t, 869
- unpack\_nas\_SLQSNasNetworkTimeCallBack\_ind\_t, 870
- unpack\_nas\_SLQSNasRFBandInfoCallback\_ind\_t, 871
- unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, 872
- unpack\_nas\_SLQSNASSwiGetChannelLock\_t, 873
- unpack\_nas\_SLQSNasSwiModemStatus\_t, 874
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t, 875
- unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 875
- unpack\_nas\_SLQSPerformNetworkScanV2\_t, 877
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t, 878
- unpack\_nas\_SLQSSwiGetHDRPersonality\_t, 879
- unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, 880
- unpack\_nas\_SLQSSwiGetHRPDStats\_t, 881
- unpack\_nas\_SLQSSwiGetLteCQI\_t, 882
- unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t, 882
- unpack\_nas\_SLQSSwiHDRPersonalityCallback\_ind\_t, 883
- unpack\_nas\_SLQSSwiNetworkDebug\_t, 884
- unpack\_nas\_SLQSSwiRandIndicatorCallback\_ind\_t, 884
- unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 888
- unpack\_pds\_GetPDSDefaults\_t, 893
- unpack\_pds\_GetPDSSState\_t, 894
- unpack\_pds\_GetPortAutomaticTracking\_t, 894
- unpack\_pds\_GetServiceAutomaticTracking\_t, 895
- unpack\_pds\_GetXTRAAutomaticDownload\_t, 896
- unpack\_pds\_GetXTRANetwork\_t, 896
- unpack\_pds\_GetXTRAValidity\_t, 897
- unpack\_pds\_SetEventReport\_ind\_t, 899
- unpack\_pds\_SetPdsState\_ind\_t, 900
- unpack\_pds\_SLQSGetAGPSCfg\_t, 901
- unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- unpack\_qos\_BindDataPort\_t, 906
- unpack\_qos\_SLQSQosGetNetworkStatus\_t, 912
- unpack\_qos\_SLQSQosSwiReadApnExtraParams\_t, 913
- unpack\_qos\_SLQSQosSwiReadDataStats\_t, 915
- unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, 916
- unpack\_qos\_SLQSSetQosEventCallback\_t, 916
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, 917
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, 917
- unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 919
- unpack\_result\_t, 928
- unpack\_rms\_GetSMSWake\_t, 929
- unpack\_rms\_SetSMSWake\_t, 930
- unpack\_RMTransferStatistics\_ind\_t, 930
- unpack\_sar\_SLQSGetRfSarState\_t, 931
- unpack\_sms\_GetSMSCAddress\_t, 932
- unpack\_sms\_SaveSMS\_t, 933
- unpack\_sms\_SendSMS\_t, 933



- unpack\_sms\_SetNewSMSCallback\_ind\_t, [935](#)
- unpack\_sms\_SetNewSMSCallback\_t, [935](#)
- unpack\_sms\_SetSMSCAddress\_t, [936](#)
- unpack\_sms\_SLQSDelSms\_t, [936](#)
- unpack\_sms\_SLQSGetIndicationRegister\_t, [937](#)
- unpack\_sms\_SLQSGetMessageWaiting\_t, [937](#)
- unpack\_sms\_SLQSGetSMS\_t, [938](#)
- unpack\_sms\_SLQSGetSmsBroadcastConfig\_t, [939](#)
- unpack\_sms\_SLQSGetSMSList\_t, [940](#)
- unpack\_sms\_SLQSGetTransLayerInfo\_t, [940](#)
- unpack\_sms\_SLQSGetTransNWRegInfo\_t, [941](#)
- unpack\_sms\_SLQSModifySMSStatus\_t, [941](#)
- unpack\_sms\_SLQSNWRegInfoCallback\_ind\_t, [942](#)
- unpack\_sms\_SLQSSendAsyncSMS\_t, [942](#)
- unpack\_sms\_SLQSSetIndicationRegister\_t, [943](#)
- unpack\_sms\_SLQSSetSmsBroadcastActivation\_t, [943](#)
- unpack\_sms\_SLQSSetSmsBroadcastConfig\_t, [944](#)
- unpack\_sms\_SLQSSetSmsStorage\_t, [944](#)
- unpack\_sms\_SLQSSmsGetMaxStorageSize\_t, [944](#)
- unpack\_sms\_SLQSSmsGetMessageProtocol\_t, [945](#)
- unpack\_sms\_SLQSSmsSetRoutes\_t, [945](#)
- unpack\_sms\_SLQSSwiGetSMSStorage\_t, [946](#)
- unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t, [947](#)
- unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t, [949](#)
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, [950](#)
- unpack\_sms\_SLQSWmsMessageWaitingCallBack\_ind\_t, [950](#)
- unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, [952](#)
- unpack\_swiaudio\_SLQSGetM2MAudioVolume\_t, [952](#)
- unpack\_swiaudio\_SLQSGetM2MAVMute\_t, [953](#)
- unpack\_swiaudio\_SLQSGetM2MSpkrGain\_t, [954](#)
- unpack\_swiavms\_SLQSAVMSEventReportInd\_t, [955](#)
- unpack\_swiavms\_SLQSAVMSSetSettings\_t, [958](#)
- unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, [960](#)
- unpack\_swiavms\_SLQSAVMSSendSelection\_t, [961](#)
- unpack\_swiavms\_SLQSAVMSSessionGetInfo\_t, [961](#)
- unpack\_swiavms\_SLQSAvmsSetEventReport\_t, [962](#)
- unpack\_swiavms\_SLQSAVMSSetSettings\_t, [962](#)
- unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, [963](#)
- unpack\_swiavms\_SLQSAVMSStartSession\_t, [963](#)
- unpack\_swiavms\_SLQSAVMSStopSession\_t, [964](#)
- unpack\_swidms\_SLQSSwiDmsGetHWWatchdog\_t, [964](#)
- unpack\_swidms\_SLQSSwiDmsGetMTU\_t, [965](#)
- unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, [966](#)
- unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t, [967](#)
- unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t, [967](#)
- unpack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, [968](#)
- unpack\_swidms\_SLQSSwiDmsSetMTU\_t, [968](#)
- unpack\_swidms\_SLQSSwiDmsSetUsbComp\_t, [969](#)
- unpack\_swiloc\_SwiLocGetAutoStart\_t, [971](#)
- unpack\_swioma\_SLQSOMADMAAlertCallback\_ind\_t, [972](#)
- unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, [975](#)
- unpack\_swioma\_SLQSOMADMGetSessionInfoExt\_t, [978](#)
- unpack\_swioma\_SLQSOMADMGetSettings\_t, [980](#)
- unpack\_swioma\_SLQSOMADMStartSession\_t, [981](#)
- unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, [981](#)
- unpack\_tmd\_SLQSTmdGetMitigationDevList\_t, [982](#)
- unpack\_tmd\_SLQSTmdGetMitigationLvl\_t, [983](#)
- unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t, [983](#)
- unpack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, [984](#)
- unpack\_uim\_ChangePin\_t, [985](#)
- unpack\_uim\_GetCardStatus\_t, [985](#)
- unpack\_uim\_GetCardStatusV2\_t, [986](#)
- unpack\_uim\_ReadTransparent\_t, [987](#)
- unpack\_uim\_SetPinProtection\_t, [988](#)
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, [989](#)
- unpack\_uim\_SLQSUIMAuthenticate\_t, [990](#)
- unpack\_uim\_SLQSUIMDepersonalization\_t, [990](#)
- unpack\_uim\_SLQSUIMEventRegister\_t, [991](#)
- unpack\_uim\_SLQSUIMGetConfiguration\_t, [992](#)
- unpack\_uim\_SLQSUIMGetFileAttributes\_t, [993](#)
- unpack\_uim\_SLQSUIMGetServiceStatus\_t, [993](#)
- unpack\_uim\_SLQSUIMGetSlotsStatus\_t, [994](#)
- unpack\_uim\_SLQSUIMGetSlotsStatusV2\_t, [995](#)
- unpack\_uim\_SLQSUIMReadRecord\_t, [996](#)
- unpack\_uim\_SLQSUIMRefreshCallback\_Ind\_t, [996](#)
- unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t, [997](#)
- unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t, [997](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind\_t, [997](#)
- unpack\_uim\_SLQSUIMWriteRecord\_t, [998](#)
- unpack\_uim\_SLQSUIMWriteTransparent\_t, [999](#)
- unpack\_uim\_UnblockPin\_t, [1000](#)

- unpack\_uim\_UnblockPinV2\_t, 1001
- unpack\_uim\_VerifyPin\_t, 1001
- unpack\_voice\_allCallStatusCallback\_ind\_t, 1003
- unpack\_voice\_DTMFEventCallback\_ind\_t, 1004
- unpack\_voice\_OTASPStatusCallback\_ind\_t, 1006
- unpack\_voice\_SLQSOriinateUSSD\_t, 1009
- unpack\_voice\_SLQSVoiceAnswerCall\_t, 1010
- unpack\_voice\_SLQSVoiceBurstDTMF\_t, 1010
- unpack\_voice\_SLQSVoiceDialCall\_t, 1011
- unpack\_voice\_SLQSVoiceEndCall\_t, 1012
- unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, 1015
- unpack\_voice\_SLQSVoiceGetCallBarring\_t, 1017
- unpack\_voice\_SLQSVoiceGetCallForwarding-  
Status\_t, 1018
- unpack\_voice\_SLQSVoiceGetCallInfo\_t, 1021
- unpack\_voice\_SLQSVoiceGetCallWaiting\_t, 1023
- unpack\_voice\_SLQSVoiceGetCLIP\_t, 1024
- unpack\_voice\_SLQSVoiceGetCLIR\_t, 1026
- unpack\_voice\_SLQSVoiceGetCNAP\_t, 1027
- unpack\_voice\_SLQSVoiceGetCOLP\_t, 1029
- unpack\_voice\_SLQSVoiceGetCOLR\_t, 1030
- unpack\_voice\_SLQSVoiceGetConfig\_t, 1032
- unpack\_voice\_SLQSVoiceManageCalls\_t, 1033
- unpack\_voice\_SLQSVoiceSendFlash\_t, 1033
- unpack\_voice\_SLQSVoiceSetCallBarringPassword-  
\_t, 1035
- unpack\_voice\_SLQSVoiceSetConfig\_t, 1036
- unpack\_voice\_SLQSVoiceSetSUPSService\_t,  
1038
- unpack\_voice\_SLQSVoiceStartContDTMF\_t, 1038
- unpack\_voice\_SLQSVoiceStopContDTMF\_t, 1039
- unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t,  
1041
- unpack\_voice\_SUPSNotificationCallback\_ind\_t,  
1044
- unpack\_voice\_USSDNotificationCallback\_ind\_t,  
1044
- unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1047
- unpack\_voice\_voicePrivacyChangeCallback\_ind\_-  
t, 1048
- unpack\_wds\_DHCPv4ClientLease\_ind\_t, 1049
- unpack\_wds\_GetAutoconnect\_t, 1049
- unpack\_wds\_GetByteTotals\_t, 1050
- unpack\_wds\_GetConnectionRate\_t, 1051
- unpack\_wds\_GetDataBearerTechnology\_t, 1052
- unpack\_wds\_GetDefaultProfile\_t, 1055
- unpack\_wds\_GetDefaultProfileNum\_t, 1055
- unpack\_wds\_GetDefaultProfileV2\_t, 1058
- unpack\_wds\_GetDormancyState\_t, 1059
- unpack\_wds\_GetLastMobileIPError\_t, 1059
- unpack\_wds\_GetMobileIP\_t, 1060
- unpack\_wds\_GetMobileIPProfile\_t, 1062
- unpack\_wds\_GetPacketStatistics\_t, 1064
- unpack\_wds\_GetPacketStatus\_t, 1066
- unpack\_wds\_GetSessionDuration\_t, 1067
- unpack\_wds\_GetSessionDurationV2\_t, 1068
- unpack\_wds\_GetSessionState\_t, 1068
- unpack\_wds\_RMSetTransferStatistics\_t, 1069
- unpack\_wds\_SetMobileIPProfile\_t, 1069
- unpack\_wds\_SLQSCreateProfile\_t, 1070
- unpack\_wds\_SLQSDeleteProfile\_t, 1070
- unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t,  
1072
- unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1074
- unpack\_wds\_SLQSGetCurrDataSystemStat\_t,  
1074
- unpack\_wds\_SLQSGetCurrentChannelRate\_t,  
1076
- unpack\_wds\_SLQSGetDataBearerTechnology\_t,  
1077
- unpack\_wds\_SLQSGetDUNCallInfo\_t, 1080
- unpack\_wds\_SLQSGetProfileSettings\_t, 1080
- unpack\_wds\_SLQSGetProfileSettingsV2\_t, 1081
- unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- unpack\_wds\_SLQSModifyProfile\_t, 1085
- unpack\_wds\_SLQSSetIPFamilyPreference\_t,  
1085
- unpack\_wds\_SLQSSetPacketSrvStatusCallback\_-  
t, 1087
- unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t,  
1093
- unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t,  
1094
- unpack\_wds\_SLQSSGetLoopback\_t, 1095
- unpack\_wds\_SLQSSStartDataSession\_t, 1095
- unpack\_wds\_SLQSSwiProfileChangeCallback\_-  
Ind\_t, 1096
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings-  
\_t, 1098
- partNoStr  
\_lifefw\_FirmwareFileInfo, 36
- path  
uim\_fileInfo, 635
- pathLen  
uim\_fileInfo, 635
- pbAuto  
unpack\_pds\_GetPortAutomaticTracking\_t, 895  
unpack\_pds\_GetServiceAutomaticTracking\_t, 895
- pbEnabled  
unpack\_pds\_GetXTRAAutomaticDownload\_t, 896
- pbPlatform  
unpack\_dms\_GetOfflineReason\_t, 687
- pci  
nas\_cellParams, 186  
nas\_ltePCI, 243  
nas\_PhyCaAggPcellInfo, 280  
nas\_PhyCaAggScellIndType, 282  
nas\_PhyCaAggScellInfo, 284  
nas\_umtsLTENbrCell, 342  
NASPhyCaAggPcellInfo, 361  
NASPhyCaAggScellArray, 362  
NASPhyCaAggScellIndType, 364  
NASPhyCaAggScellInfo, 365
- PcoList  
LibPackPCOIDList, 106
- pcsDigit

- nas\_PlmnID, [286](#)
- pcsfFQDNAddress
  - wds\_PCSCFFQDNAddressList, [1172](#)
- PdpConnState
  - unpack\_imsa\_SLQSImsaPdpStatusCallBack\_ind\_t, [757](#)
- PdpStatusConfig
  - pack\_imsa\_SLQSRegisterIMSAIndication\_t, [408](#)
- pdpType
  - pack\_wds\_SetDefaultProfile\_t, [562](#)
- pdptype
  - unpack\_wds\_GetDefaultProfile\_t, [1055](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- pds.h, [1387](#)
  - pack\_pds\_ForceXTRADownload, [1390](#)
  - pack\_pds\_GetPDSDDefaults, [1391](#)
  - pack\_pds\_GetPDSSState, [1391](#)
  - pack\_pds\_GetPortAutomaticTracking, [1391](#)
  - pack\_pds\_GetServiceAutomaticTracking, [1392](#)
  - pack\_pds\_GetXTRAAutomaticDownload, [1392](#)
  - pack\_pds\_GetXTRANetwork, [1392](#)
  - pack\_pds\_GetXTRAValidity, [1393](#)
  - pack\_pds\_PDSInjectTimeReference, [1393](#)
  - pack\_pds\_ResetPDSDData, [1393](#)
  - pack\_pds\_SLQSGetAGPSConfig, [1396](#)
  - pack\_pds\_SLQSGetGPSSStateInfo, [1397](#)
  - pack\_pds\_SLQSPDSDeterminePosition, [1397](#)
  - pack\_pds\_SLQSPDSInjectAbsoluteTimeReference, [1397](#)
  - pack\_pds\_SLQSPDSInjectPositionData, [1398](#)
  - pack\_pds\_SLQSSetAGPSConfig, [1398](#)
  - pack\_pds\_SLQSSetPositionMethodState, [1399](#)
  - pack\_pds\_SetEventReportCallback, [1394](#)
  - pack\_pds\_SetPDSDDefaults, [1394](#)
  - pack\_pds\_SetPDSSState, [1395](#)
  - pack\_pds\_SetPortAutomaticTracking, [1395](#)
  - pack\_pds\_SetServiceAutomaticTracking, [1395](#)
  - pack\_pds\_SetXTRAAutomaticDownload, [1396](#)
  - pack\_pds\_SetXTRANetwork, [1396](#)
  - pack\_pds\_StartPDSTrackingSessionExt, [1399](#)
  - pack\_pds\_StopPDSTrackingSession, [1399](#)
  - unpack\_pds\_ForceXTRADownload, [1400](#)
  - unpack\_pds\_ForceXTRADownload\_t, [1390](#)
  - unpack\_pds\_GetPDSDDefaults, [1400](#)
  - unpack\_pds\_GetPDSSState, [1400](#)
  - unpack\_pds\_GetPortAutomaticTracking, [1401](#)
  - unpack\_pds\_GetServiceAutomaticTracking, [1401](#)
  - unpack\_pds\_GetXTRAAutomaticDownload, [1401](#)
  - unpack\_pds\_GetXTRANetwork, [1402](#)
  - unpack\_pds\_GetXTRAValidity, [1402](#)
  - unpack\_pds\_PDSInjectTimeReference, [1403](#)
  - unpack\_pds\_PDSInjectTimeReference\_t, [1390](#)
  - unpack\_pds\_ResetPDSDData, [1403](#)
  - unpack\_pds\_ResetPDSDData\_t, [1390](#)
  - unpack\_pds\_SLQSGetAGPSConfig, [1406](#)
  - unpack\_pds\_SLQSGetGPSSStateInfo, [1407](#)
  - unpack\_pds\_SLQSPDSDeterminePosition, [1407](#)
  - unpack\_pds\_SLQSPDSDeterminePosition\_t, [1390](#)
  - unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference, [1407](#)
  - unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, [1390](#)
  - unpack\_pds\_SLQSPDSInjectPositionData, [1408](#)
  - unpack\_pds\_SLQSPDSInjectPositionData\_t, [1390](#)
  - unpack\_pds\_SLQSSetAGPSConfig, [1408](#)
  - unpack\_pds\_SLQSSetAGPSConfig\_t, [1390](#)
  - unpack\_pds\_SLQSSetPositionMethodState, [1408](#)
  - unpack\_pds\_SLQSSetPositionMethodState\_t, [1390](#)
  - unpack\_pds\_SetEventReport\_Ind, [1403](#)
  - unpack\_pds\_SetEventReportCallback, [1404](#)
  - unpack\_pds\_SetEventReportCallback\_t, [1390](#)
  - unpack\_pds\_SetPDSDDefaults, [1404](#)
  - unpack\_pds\_SetPDSDDefaults\_t, [1390](#)
  - unpack\_pds\_SetPDSSState, [1404](#)
  - unpack\_pds\_SetPDSSState\_t, [1390](#)
  - unpack\_pds\_SetPdsState\_Ind, [1405](#)
  - unpack\_pds\_SetPortAutomaticTracking, [1405](#)
  - unpack\_pds\_SetPortAutomaticTracking\_t, [1390](#)
  - unpack\_pds\_SetServiceAutomaticTracking, [1405](#)
  - unpack\_pds\_SetServiceAutomaticTracking\_t, [1390](#)
  - unpack\_pds\_SetXTRAAutomaticDownload, [1406](#)
  - unpack\_pds\_SetXTRAAutomaticDownload\_t, [1390](#)
  - unpack\_pds\_SetXTRANetwork, [1406](#)
  - unpack\_pds\_SetXTRANetwork\_t, [1390](#)
  - unpack\_pds\_StartPDSTrackingSessionExt, [1409](#)
  - unpack\_pds\_StartPDSTrackingSessionExt\_t, [1390](#)
  - unpack\_pds\_StopPDSTrackingSession, [1409](#)
  - unpack\_pds\_StopPDSTrackingSession\_t, [1390](#)
- peakRate
  - unpack\_qos\_tokenBucket\_t, [927](#)
- peakThroughputClass
  - LibPackGPRSRequestedQoS, [105](#)
  - wds\_GPRSQoS, [1169](#)
- PeriodicUpdateTimer
  - unpack\_dms\_PSMCfgChange\_ind\_t, [694](#)
- periodicUpdateTimer
  - dms\_PSMPeriodicUpdateTimerTlv, [71](#)
- PeriodicUpdateTimerInd
  - dms\_PSMPeriodicUpdateTimerIndTlv, [70](#)
- PersistentLpm
  - unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, [703](#)
- PersistentTechPref
  - unpack\_nas\_GetNetworkPreference\_t, [801](#)
- persoFeature
  - appStats, [41](#)
  - uim\_appStatus, [625](#)
- persoRetries
  - appStats, [41](#)
  - uim\_appStatus, [625](#)
- persoState
  - appStats, [41](#)
  - uim\_appStatus, [625](#)
- persoUnblockRetries
  - appStats, [41](#)
  - uim\_appStatus, [625](#)

- pfailureCause
  - unpack\_voice\_SLQSOriGinateUSSD\_t, [1009](#)
- PhCtxtURI
  - ims\_PhCtxtURIInfo, [90](#)
- phase
  - nas\_rxInfo, [314](#)
- PhyCaAggPcellInfo
  - NasGetLTECphyCaInfo, [355](#)
- PhyCaAggScellArray
  - NasGetLTECphyCaInfo, [355](#)
- PhyCaAggScellIDBw
  - NasGetLTECphyCaInfo, [355](#)
- PhyCaAggScellIndType
  - NasGetLTECphyCaInfo, [355](#)
- PhyCaAggScellIndex
  - NasGetLTECphyCaInfo, [355](#)
- PhyCaAggScellInfo
  - NasGetLTECphyCaInfo, [355](#)
- PhysicalLayer
  - nas\_protocolSubtypeElement, [290](#)
- PilotEnergy
  - nas\_NetworkStatEVDO, [263](#)
- PilotPN
  - nas\_PilotSetParams, [285](#)
- PilotStrength
  - nas\_PilotSetParams, [285](#)
- PilotType
  - nas\_PilotSetParams, [285](#)
- pin1Len
  - uim\_encryptedPIN1, [631](#)
- pin1Retries
  - appStats, [42](#)
  - uim\_appStatus, [625](#)
- pin1State
  - appStats, [42](#)
  - uim\_appStatus, [625](#)
- pin1Val
  - uim\_encryptedPIN1, [631](#)
- pin2Retries
  - appStats, [42](#)
  - uim\_appStatus, [625](#)
- pin2State
  - appStats, [42](#)
  - uim\_appStatus, [625](#)
- pinID
  - uim\_changeUIMPIN, [630](#)
  - uim\_setPINProtection, [647](#)
  - uim\_unblockUIMPIN, [652](#)
  - uim\_verifyUIMPIN, [654](#)
- pinLen
  - uim\_changeUIMPIN, [630](#)
  - uim\_verifyUIMPIN, [654](#)
- pinLength
  - uim\_setPINProtection, [647](#)
- pinOperation
  - uim\_setPINProtection, [647](#)
- pinProtection
  - pack\_uim\_SetPinProtection\_t, [523](#)
- pack\_uim\_UnblockPin\_t, [535](#)
- pinVal
  - uim\_changeUIMPIN, [630](#)
  - uim\_verifyUIMPIN, [654](#)
- pinValue
  - uim\_setPINProtection, [647](#)
- pkgDate
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgDateLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgDesc
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- PkgDescLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [975](#)
- pkgDescLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- PkgDescription
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [975](#)
- pkgInstallTime
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgInstallTimeLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- PkgName
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [975](#)
- pkgName
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- PkgNameLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, [975](#)
- pkgNameLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgSize
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgVendorName
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgVendorNameLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgVersionName
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgVersionNameLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, [978](#)
- pkgver

- unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 721
- PktErrRate
  - unpack\_qos\_swiQosFlow\_t, 926
- plasmaIDString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, 392
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 724
- plmn
  - nas\_GERANInfo, 206
  - nas\_LTEInfoIntraFreq, 234
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
- PlmnLen
  - nas\_QmisNasSlqsNasPCICellInfo, 294
- polarityIncluded
  - voice\_lineCtrlInfo, 1145
- port
  - loc\_IPv4Info, 160
  - loc\_IPv6Info, 161
  - unpack\_qos\_Port\_t, 909
- posDataNmea
  - pack\_pds\_SetEventReportCallback\_t, 475
- posSrc
  - unpack\_pds\_SetEventReport\_Ind\_t, 899
- PositionDataNMEA
  - unpack\_pds\_SetEventReport\_Ind\_t, 899
- positionSrc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 418
- PowerOffMode
  - unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 703
- pprofileid3gpp
  - pack\_wds\_SLQSStartDataSession\_t, 577
- pprofileid3gpp2
  - pack\_wds\_SLQSStartDataSession\_t, 577
- prDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- prDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1098
- prPCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1099
- prPCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, 1099
- Precedence
  - unpack\_qos\_swiQosFilter\_t, 922
- precedenceClass
  - LibPackGPRSRequestedQoS, 105
  - wds\_GPRSQoS, 1169
- prefNetwork
  - unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 1075
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1093
- preference
  - pack\_pds\_SetXTRANetwork\_t, 478
- Preferred
  - nas\_QmiNas3GppNetworkInfo, 292
- prefixLen
  - unpack\_qos\_IPv6Addr\_t, 908
- presentationInd
  - voice\_ECTNum, 1141
  - voice\_remotePartyNum, 1153
- priCSCFPort
  - ims\_PCSCFPortInfo, 90
- priChA
  - nas\_CDMAChannel, 179
- priChB
  - nas\_CDMAChannel, 179
- priSize
  - unpack\_dms\_GetFirmwareRevisions\_t, 681
- priVersionStr
  - \_litefw\_FirmwareFileInfo, 36
- pridns
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- pridnsv6
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- primaryDNS
  - pack\_wds\_SetDefaultProfile\_t, 562
- PrimaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- PrimaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- primaryHA
  - unpack\_wds\_GetMobileIPPProfile\_t, 1062
- privacyPref
  - pack\_voice\_SLQSVoiceSetPreferredPrivacy\_t, 551
- priver
  - unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 721
- priversion\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 678
- Profile
  - pack\_audio\_SLQSGetAudioPathConfig\_t, 373
  - pack\_audio\_SLQSGetAudioVolTLBConfig\_t, 374
  - pack\_audio\_SLQSSetAudioPathConfig\_t, 376
  - pack\_audio\_SLQSSetAudioProfile\_t, 377
  - pack\_audio\_SLQSSetAudioVolTLBConfig\_t, 378
  - pack\_swiaudio\_SLQSGetM2MAudioVolume\_t, 498
  - pack\_swiaudio\_SLQSGetM2MAVMute\_t, 499
  - pack\_swiaudio\_SLQSGetM2MSpkrGain\_t, 499
  - pack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, 500
  - pack\_swiaudio\_SLQSSetM2MAudioProfile\_t, 501
  - pack\_swiaudio\_SLQSSetM2MAudioVolume\_t, 502
  - pack\_swiaudio\_SLQSSetM2MAVMute\_t, 503
  - pack\_swiaudio\_SLQSSetM2MSpkrGain\_t, 503
  - unpack\_audio\_SLQSGetAudioProfile\_t, 658
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, 952
- ProfileID
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- ProfileId
  - pack\_wds\_SLQSGetProfileSettings\_t, 570



- profileId
  - wds\_DHCPPProfileIdTlv, 1164
  - wds\_DHCPv4ProfileId, 1167
  - wdsDhcpv4ProfileId, 1182
- ProfileId3GPP2
  - unpack\_qos\_swiQosFlow\_t, 926
- ProfileIdTlv
  - unpack\_wds\_DHCPv4ClientLease\_ind\_t, 1049
- ProfileIndex
  - PackCreateProfileOut, 581
- profileIndex
  - pack\_wds\_SLQSDeleteProfile\_t, 567
  - wds\_ProfileIdentifier, 1174
- profileInx
  - wds\_profileChange, 1173
- profileList
  - unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1074
- ProfileName
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- ProfileTlv
  - unpack\_wds\_SLQSSwiProfileChangeCallback\_ind\_t, 1096
- ProfileType
  - pack\_wds\_SLQSGetProfileSettings\_t, 570
  - PackCreateProfileOut, 581
  - unpack\_wds\_SLQSGetProfileSettings\_t, 1080
  - unpack\_wds\_SLQSGetProfileSettingsV2\_t, 1081
- profileType
  - pack\_wds\_SetDefaultProfile\_t, 562
  - pack\_wds\_SLQSDeleteProfile\_t, 568
  - wds\_DHCPPProfileIdTlv, 1164
  - wds\_DHCPv4ProfileId, 1167
  - wds\_profileChange, 1173
  - wds\_ProfileIdentifier, 1174
  - wdsDhcpv4ProfileId, 1182
- profiletype
  - pack\_wds\_GetDefaultProfile\_t, 556
  - pack\_wds\_GetDefaultProfileV2\_t, 557
- PromptFwDownload
  - pack\_swiavms\_SLQSAVMSSetSettings\_t, 506
  - pack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, 508
- PromptFwUpdate
  - pack\_swiavms\_SLQSAVMSSetSettings\_t, 506
  - pack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, 508
- Protocol
  - unpack\_nas\_GetCDMANetworkParameters\_t, 796
- provider
  - loc\_AppProviderInfoTlv, 150
- providerLen
  - loc\_AppProviderInfoTlv, 150
- ProvisionStatus
  - voice\_CLIPResp, 1132
  - voice\_CLIRResp, 1133
  - voice\_CNAPResp, 1134
  - voice\_COLPResp, 1134
  - voice\_COLRResp, 1135
- psAttachState
  - nas\_servSystem, 319
- NASServingSystemInfo, 371
- psBarStatus
  - nas\_CallBarringSysInfo, 177
  - nas\_callBarStatus, 178
- psState
  - nas\_CommInfo, 190
- psc
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
  - nas\_wcdmaCellInfo, 345
  - nas\_WCDMASysInfo, 351
- pscValid
  - nas\_WCDMASysInfo, 351
- pscsfIPv4Addr
  - wds\_PCSCFIPv4ServerAddressList, 1173
- psetting
  - unpack\_wds\_GetAutoconnect\_t, 1049
- psid
  - pack\_wds\_SLQSStopDataSession\_t, 578
  - unpack\_wds\_SLQSStartDataSession\_t, 1095
- puk1Retries
  - appStats, 42
  - uim\_appStatus, 625
- puk2Retries
  - appStats, 42
  - uim\_appStatus, 625
- pukLen
  - uim\_unblockUIMPIN, 652
- pukVal
  - uim\_unblockUIMPIN, 652
- pukValue
  - pack\_dms\_UIMUnblockPIN\_t, 398
- pwd
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- pwdsize
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- pwrDenialTime
  - voice\_lineCtrlInfo, 1145
- QMI\_LITE\_WDS\_CURRENT\_CALL\_DB\_MASK
  - wds.h, 1616
- QMI\_LITE\_WDS\_LAST\_CALL\_DB\_MASK
  - wds.h, 1616
- QCI
  - LibPackQosClassID, 142
- QFlowState
  - unpack\_qos\_QosFlowInfo\_t, 911
- QLIC
  - nas\_DeviceConfigDetail, 197
- QMI pack/unpack (lte-qmi), 31
- qaGobiApiTableBandClasses.h, 1409
- qaGobiApiTableCallControlReturnReasons.h, 1413
- qaGobiApiTableCallEndReasons.h, 1414
- qaGobiApiTableCarrierCodes.h, 1430
- qaGobiApiTableCodingScheme.h, 1432
- qaGobiApiTableGpsCapabilityCodes.h, 1435
- qaGobiApiTablePowerModes.h, 1435
- qaGobiApiTableRadioInterfaces.h, 1436
- qaGobiApiTableRegionCodes.h, 1436

- qaGobiApiTableServiceOptions.h, [1437](#)
- qaGobiApiTableSupServiceInfoClasses.h, [1439](#)
- qaGobiApiTableSwiAudio.h, [1440](#)
- qaGobiApiTableSwiOMADMSessionStatus.h, [1440](#)
- qaGobiApiTableSwiOMADMUpdateCompleteStatus.h, [1441](#)
- qaGobiApiTableVoiceCallEndReasons.h, [1443](#)
- qm\_wds\_ds\_profile\_extended\_err\_codes
  - qmerrno.h, [1455](#)
- qmTlvResult, [587](#)
  - DeviceError, [588](#)
  - DeviceResult, [588](#)
  - TlvPresenceMask, [588](#)
  - TlvResultCode, [588](#)
- qmerrno.h
  - eQCWWAN\_ERR\_API\_MUTEX\_TIMEOUT, [1452](#)
  - eQCWWAN\_ERR\_BUFFER\_SZ, [1451](#)
  - eQCWWAN\_ERR\_CANCEL\_OP, [1452](#)
  - eQCWWAN\_ERR\_DRIVER, [1452](#)
  - eQCWWAN\_ERR\_ENUM\_BEGIN, [1451](#)
  - eQCWWAN\_ERR\_ENUM\_END, [1452](#)
  - eQCWWAN\_ERR\_FILE\_COPY, [1451](#)
  - eQCWWAN\_ERR\_FILE\_OPEN, [1451](#)
  - eQCWWAN\_ERR\_GENERAL, [1451](#)
  - eQCWWAN\_ERR\_INTERNAL, [1451](#)
  - eQCWWAN\_ERR\_INVALID\_ARG, [1451](#)
  - eQCWWAN\_ERR\_INVALID\_DEVID, [1451](#)
  - eQCWWAN\_ERR\_INVALID\_FILE, [1451](#)
  - eQCWWAN\_ERR\_INVALID\_QMI\_RSP, [1451](#)
  - eQCWWAN\_ERR\_INVALID\_XID, [1452](#)
  - eQCWWAN\_ERR\_MALFORMED\_QMI\_RSP, [1451](#)
  - eQCWWAN\_ERR\_MEMORY, [1451](#)
  - eQCWWAN\_ERR\_MULTIPLE\_DEVICES, [1452](#)
  - eQCWWAN\_ERR\_MULTIPLE\_SMS\_UNSUPPORTED, [1452](#)
  - eQCWWAN\_ERR\_NO\_CANCELABLE\_OP, [1452](#)
  - eQCWWAN\_ERR\_NO\_CONNECTION, [1451](#)
  - eQCWWAN\_ERR\_NO\_DEVICE, [1451](#)
  - eQCWWAN\_ERR\_NO\_SIGNAL, [1452](#)
  - eQCWWAN\_ERR\_NONE, [1451](#)
  - eQCWWAN\_ERR\_NULL\_TLV, [1455](#)
  - eQCWWAN\_ERR\_OFFLINE, [1452](#)
  - eQCWWAN\_ERR\_PDU\_GENERATION, [1452](#)
  - eQCWWAN\_ERR\_QMI\_ABORTED, [1452](#)
  - eQCWWAN\_ERR\_QMI\_ACCESS\_DENIED, [1454](#)
  - eQCWWAN\_ERR\_QMI\_ACK\_NOT\_SENT, [1454](#)
  - eQCWWAN\_ERR\_QMI\_ARG\_TOO\_LONG, [1452](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_FAILED, [1453](#)
  - eQCWWAN\_ERR\_QMI\_AUTHENTICATION\_LOCK, [1453](#)
  - eQCWWAN\_ERR\_QMI\_BUNDLING\_NOT\_SUPPORTED, [1454](#)
  - eQCWWAN\_ERR\_QMI\_CALL\_FAILED, [1452](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_BUSY\_RSP, [1455](#)
  - eQCWWAN\_ERR\_QMI\_CARD\_CALL\_CONTROL\_FAILED, [1454](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_END, [1455](#)
  - eQCWWAN\_ERR\_QMI\_CAT\_START, [1455](#)
  - eQCWWAN\_ERR\_QMI\_CAUSE\_CODE, [1453](#)
  - eQCWWAN\_ERR\_QMI\_CLIENT\_IDS\_EXHAUSTED, [1452](#)
  - eQCWWAN\_ERR\_QMI\_CONNECT, [1451](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_IN\_USE, [1452](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_NOT\_READY, [1453](#)
  - eQCWWAN\_ERR\_QMI\_DEVICE\_STORAGE\_FULL, [1453](#)
  - eQCWWAN\_ERR\_QMI\_DISABLED, [1453](#)
  - eQCWWAN\_ERR\_QMI\_ENCODING, [1453](#)
  - eQCWWAN\_ERR\_QMI\_ENVELOPE\_CMD\_FAILURE, [1455](#)
  - eQCWWAN\_ERR\_QMI\_EVENT\_REG\_FAILED, [1455](#)
  - eQCWWAN\_ERR\_QMI\_EXTENDED\_INTERNAL, [1454](#)
  - eQCWWAN\_ERR\_QMI\_FDN\_RESTRICT, [1454](#)
  - eQCWWAN\_ERR\_QMI\_FLOW\_SUSPENDED, [1453](#)
  - eQCWWAN\_ERR\_QMI\_GENERAL, [1453](#)
  - eQCWWAN\_ERR\_QMI\_HARDWARE\_RESTRICTED, [1454](#)
  - eQCWWAN\_ERR\_QMI\_IFACE, [1451](#)
  - eQCWWAN\_ERR\_QMI\_INCOMPATIBLE\_STATE, [1454](#)
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_FLOW\_FILTER, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INCORRECT\_PIN, [1452](#)
  - eQCWWAN\_ERR\_QMI\_INFO\_UNAVAILABLE, [1454](#)
  - eQCWWAN\_ERR\_QMI\_INJECT\_TIMEOUT, [1454](#)
  - eQCWWAN\_ERR\_QMI\_INSUFFICIENT\_RESOURCES, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INTERFACE\_NOT\_FOUND, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INTERNAL, [1452](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ARG, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_CLIENT\_ID, [1452](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_DATA\_FORMAT, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ENVELOPE\_CMD, [1455](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_HANDLE, [1452](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_ID, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_INDEX, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_IP\_FAMILY\_PREF, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_MCAST\_HANDLE, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_MESSAGE\_ID, [1453](#)
  - eQCWWAN\_ERR\_QMI\_INVALID\_OPERATION, [1454](#)

- eQCWWAN\_ERR\_QMI\_INVALID\_PDP\_TYPE, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PINID, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PROFILE\_TYPE, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_PS\_ATTACH\_ACTION, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QMI\_CMD, [1454](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_QOS\_ID, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_REGISTER\_ACTION, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_SERVICE\_TYPE, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TECH\_PREF, [1452](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TERMINAL\_RSP, [1455](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TRANSITION, [1453](#)
- eQCWWAN\_ERR\_QMI\_INVALID\_TX\_ID, [1452](#)
- eQCWWAN\_ERR\_QMI\_MALFORMED\_MSG, [1452](#)
- eQCWWAN\_ERR\_QMI\_MAX, [1454](#)
- eQCWWAN\_ERR\_QMI\_MAX\_MCAST\_REQUESTS\_IN\_USE, [1453](#)
- eQCWWAN\_ERR\_QMI\_MAX\_QOS\_REQUESTS\_IN\_USE, [1453](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_DELIVERY\_FAILURE, [1453](#)
- eQCWWAN\_ERR\_QMI\_MESSAGE\_NOT\_SENT, [1453](#)
- eQCWWAN\_ERR\_QMI\_MISSING\_ARG, [1452](#)
- eQCWWAN\_ERR\_QMI\_MSG\_BLOCKED, [1454](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_ABORTED, [1454](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_NOT\_READY, [1453](#)
- eQCWWAN\_ERR\_QMI\_NETWORK\_QOS\_UNAWARE, [1453](#)
- eQCWWAN\_ERR\_QMI\_NO\_EFFECT, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_ENTRY, [1453](#)
- eQCWWAN\_ERR\_QMI\_NO\_FREE\_PROFILE, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_MEMORY, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_NETWORK\_FOUND, [1452](#)
- eQCWWAN\_ERR\_QMI\_NO\_RADIO, [1454](#)
- eQCWWAN\_ERR\_QMI\_NO\_SUBSCRIPTION, [1454](#)
- eQCWWAN\_ERR\_QMI\_NO\_THRESHOLDS, [1452](#)
- eQCWWAN\_ERR\_QMI\_NOT\_A\_MCAST\_IFACE, [1453](#)
- eQCWWAN\_ERR\_QMI\_NOT\_PROVISIONED, [1452](#)
- eQCWWAN\_ERR\_QMI\_NOT\_SUPPORTED, [1454](#)
- eQCWWAN\_ERR\_QMI\_OFFSET, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_DEVICE\_UNSUPPORTED, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_NETWORK\_UNSUPPORTED, [1452](#)
- eQCWWAN\_ERR\_QMI\_OP\_PARTIAL\_FAILURE, [1454](#)
- eQCWWAN\_ERR\_QMI\_OUT\_OF\_CALL, [1452](#)
- eQCWWAN\_ERR\_QMI\_PIN\_BLOCKED, [1453](#)
- eQCWWAN\_ERR\_QMI\_PIN\_PERM\_BLOCKED, [1453](#)
- eQCWWAN\_ERR\_QMI\_POLICY\_MISMATCH, [1454](#)
- eQCWWAN\_ERR\_QMI\_REQ, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQ\_SCH, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQ\_TO, [1451](#)
- eQCWWAN\_ERR\_QMI\_REQUESTED\_NUM\_UNSUPPORTED, [1453](#)
- eQCWWAN\_ERR\_QMI\_RSP, [1451](#)
- eQCWWAN\_ERR\_QMI\_RSP\_TO, [1451](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_ORDER, [1454](#)
- eQCWWAN\_ERR\_QMI\_SEGMENT\_TOO\_LONG, [1454](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INACTIVE, [1453](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_INVALID, [1453](#)
- eQCWWAN\_ERR\_QMI\_SESSION\_OWNERSHIP, [1453](#)
- eQCWWAN\_ERR\_QMI\_SIM\_FILE\_NOT\_FOUND, [1454](#)
- eQCWWAN\_ERR\_QMI\_SIM\_NOT\_INITIALIZED, [1453](#)
- eQCWWAN\_ERR\_QMI\_SMSC\_ADDR, [1454](#)
- eQCWWAN\_ERR\_QMI\_SUPS\_FAILURE\_CAUSE, [1454](#)
- eQCWWAN\_ERR\_QMI\_TPDU\_TYPE, [1454](#)
- eQCWWAN\_ERR\_QMI\_UNABORTABLE\_TRANSACTION, [1452](#)
- eQCWWAN\_ERR\_QMI\_UNKNOWN, [1453](#)
- eQCWWAN\_ERR\_QMI\_WIDTH, [1455](#)
- eQCWWAN\_ERR\_RESET, [1452](#)
- eQCWWAN\_ERR\_SWICM\_AM\_VERS\_ERROR, [1454](#)
- eQCWWAN\_ERR\_SWICM\_CALL\_IN\_PROGRESS, [1454](#)
- eQCWWAN\_ERR\_SWICM\_END, [1455](#)
- eQCWWAN\_ERR\_SWICM\_FAILED\_TO\_KILL\_SDK\_PROCESS, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_SESSION\_ID, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V4\_SESSION\_ID, [1454](#)
- eQCWWAN\_ERR\_SWICM\_INVALID\_V6\_SESSION\_ID, [1454](#)



- eQCWWAN\_ERR\_SWICM\_NOT\_IMPLEMENTED, [1454](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_CLNT\_NOT\_SUPPORTED, [1454](#)
- eQCWWAN\_ERR\_SWICM\_QMI\_SVC\_NOT\_SUPPORTED, [1454](#)
- eQCWWAN\_ERR\_SWICM\_SM\_NO\_AVAILABLE\_SESSIONS, [1454](#)
- eQCWWAN\_ERR\_SWICM\_SOCKET\_IN\_USE, [1454](#)
- eQCWWAN\_ERR\_SWICM\_START, [1454](#)
- eQCWWAN\_ERR\_SWICM\_TIMEOUT, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6DWN, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4DWN\_V6UP, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6DWN, [1454](#)
- eQCWWAN\_ERR\_SWICM\_V4UP\_V6UP, [1454](#)
- eQCWWAN\_ERR\_SWIDCS\_APP\_DISCONNECTED, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_DEVNODE\_NOT\_FOUND, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_END, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_FILEIO\_ERR, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_IOCTL\_ERR, [1455](#)
- eQCWWAN\_ERR\_SWIDCS\_START, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_CORRUPTED\_FW\_IMAGE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_END, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FILE\_NOT\_FOUND, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FIRMWARE\_NOT\_DOWNLOADED, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_ENTER\_DOWNLOAD\_MODE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_FLASH\_COMPLETE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_INVALID\_SLOT\_INDEX, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_PREFERENCE\_MISMATCH, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SAME\_AS\_CURRENT\_ACTIVE\_IMAGE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_IMAGE\_NOT\_SIGNED, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_SECBOOT\_INVALID\_CERT\_CHAIN, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_TOO\_MANY\_FILES, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_FAIL, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_UPDATE\_SUCCESS, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_FW\_WAIT\_FOR\_REBOOT, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_CRASH\_STATE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_INVALID\_PATH, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_DIR, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_OPENING\_FILE, [1455](#)
- eQCWWAN\_ERR\_SWIIM\_START, [1455](#)
- eQCWWAN\_ERR\_SWISM\_END, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_BEARER\_DATA\_NOT\_FOUND, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_CORRUPTED, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_MSG\_LEN\_TOO\_LONG, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_SMSC\_NUM\_CORRUPTED, [1455](#)
- eQCWWAN\_ERR\_SWISMS\_START, [1455](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP2\_ERR\_INVALID\_IDENT\_FOR\_PROFILE, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ACCESS\_ERR, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_CONTEXT\_NOT\_DEFINED, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_ERR\_OUT\_OF\_PROFILES, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_INVALID\_PROFILE\_FAMILY, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_READ\_ONLY\_FLAG\_SET, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_3GPP\_VALID\_FLAG\_NOT\_SET, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_END, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_INVALID\_PROFILE\_FAMILY, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_HNDL, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_IDENT, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_OP, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_NUM, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_PROFILE\_TYPE, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_INVALID\_SUBS\_ID, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LEN\_INVALID, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_ERR\_LIB\_NOT\_INITED, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_FAIL, [1456](#)
- eWDS\_ERR\_PROFILE\_REG\_RESULT\_LIST\_END, [1456](#)
- qmerrno.h, [1449](#)
- eQCWWANError, [1451](#)
- qm\_wds\_ds\_profile\_extended\_err\_codes, [1455](#)
- qmiSmsMessageList, [586](#)
- messageIndex, [587](#)
- messageTag, [587](#)
- qmiWDSDataBearerTechnology, [587](#)

- currentNetwork, [587](#)
- ratMask, [587](#)
- soMask, [587](#)
- qmuint16
  - SwiDataTypes.h, [1517](#)
- qmulong
  - SwiDataTypes.h, [1517](#)
- qos.h, [1456](#)
  - pack\_qos\_BindDataPort, [1458](#)
  - pack\_qos\_SLQSQosGetNetworkStatus, [1458](#)
  - pack\_qos\_SLQSQosSwiReadApnExtraParams, [1459](#)
  - pack\_qos\_SLQSQosSwiReadDataStats, [1460](#)
  - pack\_qos\_SLQSSetQosEventCallback, [1460](#)
  - unpack\_qos\_BindDataPort, [1461](#)
  - unpack\_qos\_SLQSQosGetNetworkStatus, [1461](#)
  - unpack\_qos\_SLQSQosSwiReadApnExtraParams, [1462](#)
  - unpack\_qos\_SLQSQosSwiReadDataStats, [1462](#)
  - unpack\_qos\_SLQSSetQosEventCallback, [1463](#)
  - unpack\_qos\_SLQSSetQosEventCallback\_ind, [1463](#)
  - unpack\_qos\_SLQSSetQosNWStatusCallback\_ind, [1464](#)
  - unpack\_qos\_SLQSSetQosPriEventCallback\_ind, [1464](#)
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind, [1465](#)
- qos\_BindDataPortMuxID\_t, [588](#)
  - MuxID, [588](#)
- qos\_BindDataPortPeripheralEndPointID\_t, [588](#)
  - EndPointType, [589](#)
  - IfaceID, [589](#)
- qos\_BindDataPortSIODDataPort\_t, [589](#)
  - SIODDataPort, [589](#)
- qosDeliveryOrder
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMInQoS, [1179](#)
- qosFlow
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- QosFlowInfo
  - unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, [916](#)
- RAN
  - unpack\_nas\_GetServingNetwork\_t, [805](#)
- RAT
  - nas\_QmiNas3GppNetworkRAT, [293](#)
- RATMask
  - currNetworkInfo, [56](#)
  - wds\_currNetworkInfo, [1160](#)
- RBTTlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- RFBandInfoElements, [589](#)
  - activeBandClass, [590](#)
  - activeChannel, [590](#)
  - radioInterface, [590](#)
  - unpack\_nas\_GetRFInfo\_t, [802](#)
- RFTlv
  - unpack\_nas\_SetEventReportInd\_t, [810](#)
- RMAutoConnect
  - pack\_dms\_SetCustFeature\_t, [385](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
- RPCause
  - unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t, [949](#)
- RPTlv
  - NASQmiCbkNasSystemSelPrefInd, [369](#)
- RRTlv
  - unpack\_nas\_SetEventReportInd\_t, [810](#)
- RSRPThresListLen
  - nas\_RSRPThresh, [311](#)
- RSRQThresListLen
  - nas\_RSRQThresh, [312](#)
- RSSIThresListLen
  - nas\_RSSIThresh, [313](#)
- RTIDTlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- RTTlv
  - unpack\_ims\_SLQSVolPCfgCallback\_ind\_t, [753](#)
- RX\_EC\_IO
  - nas\_NetworkStat1x, [262](#)
- RX\_PWR
  - nas\_NetworkStat1x, [262](#)
  - nas\_NetworkStatEVDO, [263](#)
- RXBOTlv
  - unpack\_wds\_SLQSDUNCallInfoCallback\_ind\_t, [1072](#)
- RXChan
  - nas\_LTEInfo, [232](#)
- rXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- rXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- rXOKBytesCount
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- rXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- rXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- rXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, [1066](#)
- radio
  - unpack\_nas\_GetSignalStrengths\_t, [806](#)
- radio\_if
  - pack\_nas\_SLQSNasGetTxRxInfo\_t, [440](#)
- radioIf
  - nas\_ecioListElement, [199](#)
  - nas\_errorRateListElement, [204](#)
  - nas\_RFBandInfoExtTlv, [301](#)
  - nas\_RFBandInfoTlv, [302](#)
  - nas\_RFBandwidthInfoTlv, [305](#)
  - nas\_RFDedicatedBandInfoTlv, [307](#)
  - nas\_rsrqInformation, [311](#)
  - nas\_rxSignalStrengthListElement, [316](#)
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, [869](#)

- Radiolfaces
  - dms\_devCaps, [57](#)
  - unpack\_dms\_GetDeviceCap\_t, [669](#)
  - unpack\_dms\_GetDeviceCapabilities\_t, [671](#)
  - unpack\_nas\_GetServingNetwork\_t, [805](#)
- RadiolfacesSize
  - dms\_devCaps, [57](#)
  - unpack\_dms\_GetDeviceCap\_t, [669](#)
  - unpack\_nas\_GetServingNetwork\_t, [805](#)
- radiolfacesSize
  - unpack\_dms\_GetDeviceCapabilities\_t, [671](#)
- radiolInterface
  - nas\_RFBandInfoElements, [298](#)
  - nas\_RfBandInfoExtFormatElements, [299](#)
  - nas\_RfBandwidthInfoElements, [304](#)
  - nas\_RfDedicatedBandInfoElements, [307](#)
  - nas\_RFInfoTlv, [308](#)
  - nas\_roamIndList, [309](#)
  - nas\_servSystem, [319](#)
  - nas\_SignalStrengthTlv, [320](#)
  - nas\_timeInfo, [334](#)
  - RFBandInfoElements, [590](#)
- radiolInterfaceList
  - NASServingSystemInfo, [371](#)
- radiolInterfaceNo
  - NASServingSystemInfo, [371](#)
- radiolInterfaceSize
  - nas\_RFInfoTlv, [309](#)
- randomizationWindow
  - dms\_PSMRandomizationWindowTlv, [71](#)
- range
  - unpack\_qos\_Port\_t, [909](#)
- rankIndicatorTlv
  - unpack\_nas\_SLQSSwiRandIndicatorCallback\_Ind\_t, [884](#)
- rat
  - nas\_CSGID, [191](#)
  - nas\_MNRInfo, [256](#)
- RatDMTlv
  - NASQmiCbkNasSystemSelPrefInd, [369](#)
- RatDisabledMask
  - NASRatDisabledMaskTlv, [369](#)
- ratDisabledMask
  - nas\_RatDisabledMaskTlv, [297](#)
- RatHandover
  - unpack\_imsa\_SLQSImsaRatStatusCallBack\_ind\_t, [758](#)
- RatHandoverStatus
  - imsa\_RatHandoverStatusInfo, [100](#)
- RatHandoverStatusConfig
  - pack\_imsa\_SLQSRegisterIMSAIndication\_t, [408](#)
- ratMask
  - qmiWDSDataBearerTechnology, [587](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- rawHorConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- rawHorUncCircular
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- rawLen
  - uim\_fileAttributes, [634](#)
- rawValue
  - uim\_fileAttributes, [634](#)
- readRecord
  - pack\_uim\_SLQSUIReadRecord\_t, [529](#)
- readTransparent
  - pack\_uim\_ReadTransparent\_t, [522](#)
- Reason
  - pack\_voice\_SLQSVoiceGetCallForwardingStatus\_t, [543](#)
  - pack\_voice\_SLQSVoiceSetCallBarringPassword\_t, [550](#)
- reason
  - pack\_voice\_SLQSVoiceGetCallBarring\_t, [542](#)
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, [554](#)
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [919](#)
  - voice\_ccSUPSType, [1131](#)
  - voice\_redirNumInfo, [1151](#)
- receiptAction
  - sms\_routeEntry, [603](#)
- reconfigReqd
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, [1087](#)
- record
  - uim\_readRecordInfo, [640](#)
  - uim\_writeRecordInfo, [654](#)
- recordCount
  - uim\_fileAttributes, [634](#)
- recordSize
  - uim\_fileAttributes, [634](#)
- RedirPartyNum
  - voice\_arrRedirPartyNum, [1118](#)
- refData
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [723](#)
- refID
  - pack\_cat\_CATSendTerminalResponse\_t, [380](#)
- refString
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, [723](#)
- ReferenceID
  - cat\_AIPhIdIdentifierTlv, [49](#)
  - cat\_EventIDDataTlv, [51](#)
- refpn
  - nas\_CDMAInfo, [181](#)
- refreshComplete
  - pack\_uim\_SLQSUIRefreshComplete\_t, [529](#)
- refreshEvent
  - unpack\_uim\_SLQSUIRefreshCallback\_Ind\_t, [996](#)
- RefreshMode
  - cat\_RefreshTlv, [52](#)
- RefreshStage
  - cat\_RefreshTlv, [52](#)
- regAction
  - pack\_nas\_SLQSInitiateNetworkRegistration\_t, [434](#)
- RegForeignNID

- unpack\_nas\_GetCDMANetworkParameters\_t, 796
- RegForeignSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, 796
- RegHomeSID
  - unpack\_nas\_GetCDMANetworkParameters\_t, 797
- regInd
  - unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t, 947
- regPrd
  - nas\_AddCDMASysInfo, 173
- regRefresh
  - pack\_uim\_SLQSUIRefreshRegister\_t, 531
- regRejectInfoValid
  - nas\_GSMSysInfo, 213
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_WCDMASysInfo, 351
- regState
  - nas\_servSystem, 319
- RegStatusConfig
  - pack\_imsa\_SLQSRegisterIMSAIndication\_t, 408
- registerFlag
  - uim\_registerRefresh, 644
- RegistrationState
  - unpack\_nas\_GetServingNetwork\_t, 805
- registrationState
  - NASServingSystemInfo, 371
- rejCause
  - nas\_GSMSysInfo, 213
  - nas\_LTESysInfo, 252
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_WCDMASysInfo, 351
- rejectCause
  - nas\_NR5GSysInfoTlv, 272
  - nas\_RejectReasonTlv, 297
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, 869
- rejectSrvDomain
  - nas\_GSMSysInfo, 213
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_WCDMASysInfo, 351
- releaseDate
  - \_lifefw\_FirmwareFileInfo, 36
- reliabilityClass
  - LibPackGPRSRequestedQoS, 105
  - wds\_GPRSQoS, 1169
- remPartyNumber
  - voice\_remotePartyNum, 1153
- RemotePartyName
  - voice\_getAllCallRmtPtyName, 1143
- RemotePartyNum
  - voice\_getAllCallRmtPtyNum, 1143
- ReqMitigationLvl
  - unpack\_tmd\_SLQSTmdGetMitigationLvl\_t, 983
- resBerRatio
  - LibPackUMTSQoS, 146
  - wds\_UMTSMInQoS, 1179
- ResCode
  - unpack\_audio\_SLQSGetAudioVoITLBConfig\_t, 659
  - unpack\_audio\_SLQSSetAudioVoITLBConfig\_t, 659
  - unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 723
- resetDelay
  - pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, 509
  - swidms\_SwiDmsGetHWWatchdog, 618
- resetInfoInd
  - pack\_dms\_SLQSDmsSwiIndicationRegister\_t, 389
- resultcode
  - unpack\_swiavms\_SLQSAVMSEventReportInd\_t, 955
  - unpack\_swiavms\_SLQSAVMSSetSettings\_t, 958
  - unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, 960
  - unpack\_swiavms\_SLQSAVMSSendSelection\_t, 961
  - unpack\_swiavms\_SLQSAVMSSessionGetInfo\_t, 962
  - unpack\_swiavms\_SLQSAvmsSetEventReport\_t, 962
  - unpack\_swiavms\_SLQSAVMSSetSettings\_t, 962
  - unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, 963
  - unpack\_swiavms\_SLQSAVMSStartSession\_t, 963
  - unpack\_swiavms\_SLQSAVMSStopSession\_t, 964
- RetryCount
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, 975
- revPolarity
  - voice\_lineCtrlInfo, 1145
- revTunneling
  - unpack\_wds\_GetMobileIPProfile\_t, 1062
- ReverseMac
  - nas\_protocolSubtypeElement, 290
- rfBandInfo
  - unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, 871
- rfBandInfoExt
  - unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, 871
- rfBandwidthInfo
  - unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, 871
- rfDedicatedBandInfo
  - unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, 871
- RfSarState
  - pack\_sar\_SLQSSetRfSarState\_t, 488
- rfbandInfoList
  - unpack\_nas\_SLQSNasGetRFInfo\_t, 864
- RingBkTmr
  - ims\_RngBkTmrInfo, 91

- RingTmr
  - ims\_RngTmrInfo, 91
- RmTrasferStaticsReq
  - pack\_wds\_RMSetTransferStatistics\_t, 560
- rmTrasferStaticsReq, 590
  - bResetStatistics, 590
  - ulMask, 590
- rms.h, 1466
  - pack\_rms\_GetSMSWake, 1466
  - pack\_rms\_SetSMSWake, 1467
  - unpack\_rms\_GetSMSWake, 1467
  - unpack\_rms\_SetSMSWake, 1467
- RmtPtyNum
  - voice\_arrRemotePartyNum, 1119
- roamIndicator
  - nas\_roamIndList, 309
- RoamIndicatorVal
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 831
- roamOrigVoiceSO
  - voice\_prefVoiceSO, 1150
- RoamPref
  - nas\_RoamPrefTlv, 310
  - NASRoamPreferenceTlv, 369
- roamStatus
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_sysInfoCommon, 329
- roamStatusValid
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 275
  - nas\_sysInfoCommon, 329
- roamTimerValue
  - voice\_roamTimer, 1154
- Roaming
  - nas\_QmiNas3GppNetworkInfo, 292
  - unpack\_nas\_GetCDMANetworkParameters\_t, 797
  - unpack\_nas\_GetServingNetwork\_t, 805
- roaming
  - unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, 812
- RoamingIndicatorList
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 831
- routeList
  - sms\_setRoutesReq, 607
- routeStorage
  - sms\_routeEntry, 603
- RptIMSCapability
  - pack\_dms\_SetIndicationRegister\_t, 387
- rptPosData
  - pack\_pds\_SetEventReportCallback\_t, 475
- rptRate
  - nas\_LTESigRptCfg, 247
  - nas\_LTESigRptConfig, 248
- rscp
  - nas\_rxInfo, 314
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
  - tdscdmaSigInfoExt, 619
- rsrp
  - lteSSInfo, 170
  - nas\_cellParams, 186
  - nas\_QmisNasSlqsNasPCInfo, 295
  - nas\_rxInfo, 314
  - nas\_RxSigInfo, 315
  - nas\_umtsLTENbrCell, 342
- rsrpRx0
  - nas\_QmisNasSlqsNasPCInfo, 295
- rsrpRx1
  - nas\_QmisNasSlqsNasPCInfo, 295
- rsrpIlevel
  - nas\_lteRsrpInformation, 244
- rsrq
  - lteSSInfo, 170
  - nas\_cellParams, 186
  - nas\_QmisNasSlqsNasPCInfo, 296
  - nas\_rsrqInformation, 311
  - nas\_SccRxInfo, 317
  - nas\_umtsLTENbrCell, 342
- rsrqDelta
  - nas\_SLQSSignalStrengthsIndReq, 322
- rsrqInfo
  - nas\_SLQSSignalStrengthsInformation, 324
  - unpack\_nas\_SLQSGetSignalStrength\_t, 833
- rsrqRx0
  - nas\_QmisNasSlqsNasPCInfo, 296
- rsrqRx1
  - nas\_QmisNasSlqsNasPCInfo, 296
- rssi
  - cdmaSSInfo, 53
  - hdrSSInfo, 83
  - lteSSInfo, 170
  - nas\_cellParams, 186
  - nas\_gsmCellInfo, 209
  - tdscdmaSigInfoExt, 619
  - unpack\_nas\_GetSignalStrengths\_t, 806
- rx\_bytes
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1093
- rx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1093
- RxByteOKCnt
  - wds\_RXBytesOKTlv, 1175
- rxChainIndex
  - nas\_RxSigInfo, 315
- RxDropConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, 930
- rxLev
  - nas\_GERANInfo, 206
- rxOKBytesCount
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, 1080
- RxOkByteCountTlv
  - unpack\_RMTransferStatistics\_ind\_t, 930
- RxOkConutTlv

- unpack\_RMTransferStatistics\_ind\_t, 931
- rxPower
  - nas\_rxInfo, 314
  - nas\_RxSigInfo, 315
- RxQFilter
  - unpack\_qos\_QosFlowInfo\_t, 911
- RxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, 911
- rxSignalStrength
  - nas\_rxSignalStrengthListElement, 316
- rxSignalStrengthDelta
  - nas\_SLQSSignalStrengthsIndReq, 322
- rxSignalStrengthInfo
  - nas\_SLQSSignalStrengthsInformation, 324
- rxSignalStrengthList
  - unpack\_nas\_SLQSGetSignalStrength\_t, 833
- rxSignalStrengthListLen
  - unpack\_nas\_SLQSGetSignalStrength\_t, 834
- SCI
  - unpack\_nas\_GetCDMANetworkParameters\_t, 797
- SCM
  - unpack\_nas\_GetCDMANetworkParameters\_t, 797
- SCTlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, 749
- SDPTlv
  - NASQmiCbkNasSystemSelPrefInd, 369
- SDTlv
  - unpack\_ims\_SLQSVolPCfgCallBack\_ind\_t, 753
- SDU\_HDR\_LEN
  - common.h, 1195
- SFTlv
  - unpack\_ims\_SLQSSMSCfgCallBack\_ind\_t, 750
- SHORT
  - SwiDataTypes.h, 1517
- SI
  - voice\_calledPartyInfo, 1122
  - voice\_callFWExtInfo, 1127
  - voice\_callingPartyInfo, 1131
  - voice\_redirNumInfo, 1151
- SINTlv
  - unpack\_ims\_SLQSSMSCfgCallBack\_ind\_t, 750
- SIODataPort
  - qos\_BindDataPortSIODataPort\_t, 589
- SIPLocalPort
  - ims\_SIPPortInfo, 93
- sIntraSearch
  - nas\_LTEInfoIntraFreq, 234
- SLQSSSTlv
  - unpack\_nas\_SetEventReportInd\_t, 810
- SMS\_CONFIG\_LEN
  - sms.h, 1473
- SMS\_MAX\_SMS\_ROUTES
  - sms.h, 1474
- SMS\_NUM\_OF\_SET
  - sms.h, 1474
- SMSC\_TYPE\_LEN
  - sms.h, 1474
- sMSCAddressInfo, 608
  - data, 608
  - length, 608
- sMSCAddressTlv, 608
  - SMSCInfo, 608
  - TlvPresent, 609
- SMSCInfo
  - sMSCAddressTlv, 608
- SMSCSTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 935
- sMSEtwsMessageInfo, 609
  - data, 609
  - length, 609
  - notificationType, 609
- sMSEtwsMessageTlv, 609
  - EtwsMessageInfo, 610
  - TlvPresent, 610
- sMSEtwsPImnInfo, 610
  - mobileCountryCode, 610
  - mobileNetworkCode, 610
- sMSMTMessageInfo, 611
  - messageIndex, 611
  - storageType, 611
- sMSMessageModelInfo, 610
  - messageMode, 611
- sMSOnIMSInfo, 611
  - smsOnIMS, 611
- sMSOnIMSTlv, 612
  - IMSInfo, 612
  - TlvPresent, 612
- SMSSupport
  - pack\_dms\_SetCustFeature\_t, 385
  - unpack\_dms\_GetCustFeature\_t, 667
- sMSTransferRouteMTMessageInfo, 612
  - ackIndicator, 613
  - data, 613
  - format, 613
  - length, 613
  - transactionID, 613
- SNR
  - nas\_NetworkStatEVDO, 263
- sNonIntraSearch
  - nas\_LTEInfoIntraFreq, 234
- sNumScellsConfig
  - unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t, 811
- SO
  - nas\_NetworkStat1x, 262
- SOMask
  - currNetworkInfo, 56
  - wds\_currNetworkInfo, 1160
- SPC\_SIZE
  - dms.h, 1208
- SPTlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, 749
- sPhyCaAggDIBW
  - unpack\_nas\_SetNasLTECphyCalndCallback\_ind\_t, 811
- sPhyCaAggPcellInfo



- unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [811](#)
- sPhyCaAggScellIDBw
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [811](#)
- sPhyCaAggScellIndType
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [811](#)
- sPhyCaAggScellIndex
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [811](#)
- sPhyCaAggScellInfo
  - unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, [811](#)
- SRTlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, [749](#)
- SSInfo
  - unpack\_nas\_SetServingSystemCallback\_ind\_t, [813](#)
- sLQSSignalStrengthsInfo
  - nas\_SLQSSignalStrengthsTlv, [324](#)
- SSTlv
  - unpack\_nas\_SetEventReportInd\_t, [810](#)
- STTlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, [749](#)
- SUPSIInformation
  - unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, [1042](#)
- SUPSType
  - pack\_voice\_SLQSVoiceManageCalls\_t, [548](#)
- SWI\_API
  - SwiDataTypes.h, [1517](#)
- SWIWWANCMAPI.h, [1545](#)
- samplesPerBatch
  - loc\_accelAcceptReady, [148](#)
  - loc\_accelTempAcceptReady, [149](#)
  - loc\_gyroAcceptReady, [157](#)
  - loc\_gyroTempAcceptReady, [157](#)
- sar.h, [1468](#)
  - pack\_sar\_SLQSGetRfSarState, [1468](#)
  - pack\_sar\_SLQSSetRfSarState, [1468](#)
  - unpack\_sar\_SLQSGetRfSarState, [1469](#)
  - unpack\_sar\_SLQSSetRfSarState, [1469](#)
  - unpack\_sar\_SLQSSetRfSarState\_t, [1468](#)
- sbas\_almanac\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- sbas\_ephemeris\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- sbas\_health\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- sbas\_visible\_sv\_msk
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- scell\_idx
  - nas\_PhyCaAggScellIndex, [281](#)
  - NASPhyCaAggScellArray, [362](#)
  - NASPhyCaAggScellIndex, [363](#)
- scell\_state
  - nas\_PhyCaAggScellIndType, [282](#)
- nas\_PhyCaAggScellInfo, [284](#)
- NASPhyCaAggScellArray, [362](#)
- NASPhyCaAggScellIndType, [364](#)
- NASPhyCaAggScellInfo, [365](#)
- scrAmrEnable
  - ims\_EnabSCRAMRInfo, [88](#)
- scrAmrWbEnable
  - ims\_EnabSCRAMRWBInfo, [88](#)
- screeningInd
  - voice\_connectNumInfo, [1137](#)
- sduErrorRatio
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMInQoS, [1179](#)
- seDNSIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1099](#)
- seDNSIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1099](#)
- sePCSCFIPv4Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1099](#)
- sePCSCFIPv6Address
  - unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings\_t, [1099](#)
- secActivate
  - uim\_fileAttributes, [634](#)
- secActivateMask
  - uim\_fileAttributes, [634](#)
- secChA
  - nas\_CDMAChannel, [179](#)
- secChB
  - nas\_CDMAChannel, [179](#)
- secDeactivate
  - uim\_fileAttributes, [634](#)
- secDeactivateMask
  - uim\_fileAttributes, [634](#)
- secIncrease
  - uim\_fileAttributes, [634](#)
- secIncreaseMask
  - uim\_fileAttributes, [634](#)
- SecProt
  - nas\_protocolSubtypeElement, [290](#)
- secRead
  - uim\_fileAttributes, [634](#)
- secReadMask
  - uim\_fileAttributes, [634](#)
- secWrite
  - uim\_fileAttributes, [635](#)
- secWriteMask
  - uim\_fileAttributes, [635](#)
- secdns
  - unpack\_wds\_GetDefaultProfile\_t, [1055](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- secdnsv6
  - unpack\_wds\_GetDefaultProfile\_t, [1055](#)
  - unpack\_wds\_GetDefaultProfileV2\_t, [1058](#)
- second

- nas\_timeInfo, 334
- nas\_UniversalTime, 343
- secondaryDNS
  - pack\_wds\_SetDefaultProfile\_t, 562
- SecondaryDNSV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- SecondaryDNSV6
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- secondaryHA
  - unpack\_wds\_GetMobileIPProfile\_t, 1062
- SectorIDLen
  - nas\_NetworkStatEVDO, 263
- secureBootEnabled
  - unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, 966
- selNetwork
  - nas\_servSystem, 319
- selected
  - sms\_BroadcastConfig, 595
  - sms\_CDMABroadcastConfig, 596
- selectedNetwork
  - NASServingSystemInfo, 371
- selection
  - pack\_swiaVms\_SLQSAVMSSendSelection\_t, 504
  - pack\_swioMa\_SLQSOMADMSendSelection\_t, 515
  - pack\_swioMa\_SLQSOMADMSendSelectionExt\_t, 516
- sendStatus
  - unpack\_sms\_SLQSWmsAsyncRawSendCallback\_ind\_t, 949
- sensorData\_t, 590
  - flags, 591
  - sensorDataLen, 592
  - timeOfFirstSample, 592
  - timeOffset, 592
  - xAxis, 592
  - yAxis, 592
  - zAxis, 592
- sensorDataLen
  - sensorData\_t, 592
- ServerAddrList
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- serverAddrTypeMask
  - pack\_loc\_SLQSLOCGetServer\_t, 413
- serverStatus
  - unpack\_loc\_GetServer\_Ind\_t, 775
  - unpack\_loc\_SetServer\_Ind\_t, 791
- serverType
  - pack\_loc\_SLQSLOCGetServer\_t, 413
  - pack\_loc\_SLQSLOCSetServer\_t, 422
  - unpack\_loc\_GetServer\_Ind\_t, 775
- serviceCategory
  - sms\_CDMABroadcastConfig, 596
- serviceDomain
  - nas\_RejectReasonTlv, 297
  - unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t, 869
- ServiceStatusConfig
  - pack\_imsa\_SLQSRegisterIMSAlndication\_t, 408
- servingCellId
  - nas\_LTEInfoIntraFreq, 234
- ServingSystem
  - unpack\_nas\_SLQSGetServingSystem\_t, 826
  - unpack\_nas\_SLQSGetServingSystemV2\_t, 831
- sessExp
  - ims\_SessDurInfo, 92
- sessionControl
  - pack\_pds\_StartPDSTrackingSessionExt\_t, 484
- sessionEndReason
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 1087
- SessionId
  - pack\_loc\_Start\_t, 424
  - pack\_loc\_Stop\_t, 424
- sessionId
  - unpack\_loc\_PositionRpt\_Ind\_t, 787
- sessionInfo
  - pack\_uim\_ChangePin\_t, 521
  - pack\_uim\_ReadTransparent\_t, 522
  - pack\_uim\_SetPinProtection\_t, 523
  - pack\_uim\_SLQSUIMAuthenticate\_t, 524
  - pack\_uim\_SLQSUIMGetFileAttributes\_t, 526
  - pack\_uim\_SLQSUIMGetServiceStatus\_t, 527
  - pack\_uim\_SLQSUIMReadRecord\_t, 529
  - pack\_uim\_SLQSUIMRefreshComplete\_t, 529
  - pack\_uim\_SLQSUIMRefreshGetLastEvent\_t, 530
  - pack\_uim\_SLQSUIMRefreshOK\_t, 530
  - pack\_uim\_SLQSUIMRefreshRegister\_t, 531
  - pack\_uim\_SLQSUIMSetServiceStatus\_t, 531
  - pack\_uim\_SLQSUIMWriteRecord\_t, 533
  - pack\_uim\_SLQSUIMWriteTransparent\_t, 534
  - pack\_uim\_UnblockPin\_t, 535
  - pack\_uim\_VerifyPin\_t, 536
- SessionInfoConfig
  - unpack\_swioMa\_SLQSOMADMAAlertCallback\_ind\_t, 972
- SessionInfoFota
  - unpack\_swioMa\_SLQSOMADMAAlertCallback\_ind\_t, 972
- SessionInfoNotification
  - unpack\_swioMa\_SLQSOMADMAAlertCallback\_ind\_t, 972
- sessionOperation
  - pack\_pds\_StartPDSTrackingSessionExt\_t, 484
- sessionResponse
  - unpack\_swiaVms\_SLQSAVMSStartSession\_t, 964
- sessionServerOption
  - pack\_pds\_StartPDSTrackingSessionExt\_t, 484
- SessionState
  - unpack\_swioMa\_SLQSOMADMGetSessionInfo\_t, 975
- sessionState
  - unpack\_swioMa\_SLQSOMADMGetSessionInfoExt\_t, 978
- SessionStatus



- unpack\_pds\_SetEventReport\_Ind\_t, 899
- sessionStatus
  - unpack\_loc\_PositionRpt\_Ind\_t, 787
  - unpack\_omaDmNotificationsTlv\_t, 892
- SessionType
  - pack\_swioama\_SLQSOMADMGetSessionInfo\_t, 515
  - unpack\_swioama\_SLQSOMADMGetSessionInfo\_t, 975
- sessionType
  - pack\_pds\_StartPDSTrackingSessionExt\_t, 484
  - pack\_swiaavms\_SLQSAVMSStartSession\_t, 508
  - pack\_swiaavms\_SLQSAVMSStopSession\_t, 509
  - pack\_swioama\_SLQSOMADMCancelSession\_t, 514
  - pack\_swioama\_SLQSOMADMCancelSessionExt\_t, 514
  - pack\_swioama\_SLQSOMADMStartSession\_t, 518
  - pack\_swioama\_SLQSOMADMStartSessionExt\_t, 519
  - uim\_refreshevent, 644
  - uim\_sessionInformation, 646
  - uim\_UIMSessionInformation, 652
  - unpack\_omaDmFotaTlv\_t, 891
- set\_fix\_rate
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
- set\_fix\_type
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
- set\_function
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
- set\_max\_dist
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
- set\_max\_time
  - pack\_swiloc\_SwiLocSetAutoStart\_t, 513
- SetupEventList
  - cat\_EventListTlv, 52
- Severity
  - unpack\_swioama\_SLQSOMADMGetSessionInfo\_t, 975
- severity
  - unpack\_omaDmFotaTlv\_t, 891
- shortName
  - nas\_PLMNNetworkNameData, 288
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- shortNameCI
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- shortNameEn
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- shortNameLen
  - nas\_PLMNNetworkNameData, 288
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- shortNameSB
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- shortNameSpareBits
  - nas\_PLMNNetworkNameData, 288
- sid
  - nas\_CDMAInfo, 181
  - nas\_sidNid, 319
- unpack\_nas\_GetHomeNetwork\_t, 800
- unpack\_nas\_SLQSGetHomeNetwork\_t, 816
- SidNid
  - nas\_homeSIDNID, 221
- SigCompEn
  - ims\_SigCompEnInfo, 93
- SigInd
  - LibPackUMTSReqQoSsigInd, 147
- sigInfo
  - nas\_SccRxInfo, 317
- signal
  - voice\_signalInfo, 1154
- signalStrength
  - nas\_SignalStrengthTlv, 320
- signalStrengthReqMask
  - unpack\_nas\_SLQSGetSignalStrength\_t, 834
- signalType
  - voice\_signalInfo, 1154
- simBusy
  - uim\_simBusyStatus, 647
- simBusyLength
  - uim\_simBusyStatus, 647
- SimCapability
  - dms\_devCaps, 57
  - unpack\_dms\_GetDeviceCap\_t, 669
- simCapability
  - unpack\_dms\_GetDeviceCapabilities\_t, 671
- SimRejInfo
  - nas\_SimRejInfoTlv, 321
- SimVoiceDataCap
  - dms\_devSubsVoiceDataList, 64
- sinr
  - hdrSSInfo, 83
  - nas\_SLQSSignalStrengthsInformation, 324
  - tdscdmaSigInfoExt, 619
  - unpack\_nas\_SLQSGetSignalStrength\_t, 834
- sinrDelta
  - nas\_SLQSSignalStrengthsIndReq, 322
- sinrThresholdList
  - nas\_SLQSSignalStrengthsIndReq, 322
- sinrThresholdListLen
  - nas\_SLQSSignalStrengthsIndReq, 322
- sku\_str
  - unpack\_dms\_GetFirmwareInfo\_t, 678
- skuStr
  - \_litefw\_FirmwareFileInfo, 36
- slot
  - pack\_uim\_SLQSUIMPowerDown\_t, 527
  - pack\_uim\_SLQSUIMPowerUp\_t, 528
- slot\_t, 592
  - bICCID, 593
  - bICCIDLength, 593
  - bLogicalSlot, 593
  - uPhyCardStatus, 593
  - uPhySlotStatus, 593
- slotInf, 593
  - AppStatus, 594
  - cardState, 594

- errorState, [594](#)
- numApp, [594](#)
- upinRetries, [594](#)
- upinState, [594](#)
- upukRetries, [594](#)
- SlotInfo
  - uim\_cardStatus, [629](#)
- slots\_t, [594](#)
  - uimSlotStatus, [595](#)
- slotsstatusChange
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, [989](#)
- SlqsProfile3GPP
  - unpackWdsProfileParam, [1109](#)
  - unpackWdsProfileParamV2, [1110](#)
  - wds\_profileInfo, [1174](#)
- SlqsProfile3GPP2
  - unpackWdsProfileParam, [1109](#)
  - unpackWdsProfileParamV2, [1110](#)
  - wds\_profileInfo, [1174](#)
- sms.h
  - LITEQMI\_QMI\_CBK\_PARAM\_NOCHANGE, [1474](#)
  - LITEQMI\_QMI\_CBK\_PARAM\_RESET, [1474](#)
  - LITEQMI\_QMI\_CBK\_PARAM\_SET, [1474](#)
- sms.h, [1470](#)
  - eqmiCbKsetStatus, [1474](#)
  - MAX\_MSE\_TWS\_MSG, [1473](#)
  - MAX\_SMS\_LIST\_SIZE, [1473](#)
  - pack\_sms\_GetSMSCAddress, [1474](#)
  - pack\_sms\_SLQSDDeleteSMS, [1476](#)
  - pack\_sms\_SLQSGetIndicationRegister, [1476](#)
  - pack\_sms\_SLQSGetMessageWaiting, [1476](#)
  - pack\_sms\_SLQSGetSMS, [1477](#)
  - pack\_sms\_SLQSGetSMSList, [1477](#)
  - pack\_sms\_SLQSGetSmsBroadcastConfig, [1477](#)
  - pack\_sms\_SLQSGetTransLayerInfo, [1478](#)
  - pack\_sms\_SLQSGetTransNWRRegInfo, [1478](#)
  - pack\_sms\_SLQSModifySMSStatus, [1478](#)
  - pack\_sms\_SLQSSendAsyncSMS, [1479](#)
  - pack\_sms\_SLQSSetIndicationRegister, [1479](#)
  - pack\_sms\_SLQSSetSmsBroadcastActivation, [1479](#)
  - pack\_sms\_SLQSSetSmsBroadcastConfig, [1480](#)
  - pack\_sms\_SLQSSetSmsStorage, [1480](#)
  - pack\_sms\_SLQSSmsGetMaxStorageSize, [1481](#)
  - pack\_sms\_SLQSSmsGetMessageProtocol, [1481](#)
  - pack\_sms\_SLQSSmsSetRoutes, [1481](#)
  - pack\_sms\_SLQSSwiGetSMSStorage, [1482](#)
  - pack\_sms\_SaveSMS, [1474](#)
  - pack\_sms\_SendSMS, [1475](#)
  - pack\_sms\_SetNewSMSCallback, [1475](#)
  - pack\_sms\_SetSMSCAddress, [1475](#)
  - SMS\_CONFIG\_LEN, [1473](#)
  - SMS\_MAX\_SMS\_ROUTES, [1474](#)
  - SMS\_NUM\_OF\_SET, [1474](#)
  - SMSC\_TYPE\_LEN, [1474](#)
  - unpack\_sms\_GetSMSCAddress, [1482](#)
  - unpack\_sms\_SLQSDDeleteSMS, [1484](#)
  - unpack\_sms\_SLQSGetIndicationRegister, [1484](#)
  - unpack\_sms\_SLQSGetMessageWaiting, [1485](#)
  - unpack\_sms\_SLQSGetSMS, [1485](#)
  - unpack\_sms\_SLQSGetSMSList, [1486](#)
  - unpack\_sms\_SLQSGetSmsBroadcastConfig, [1485](#)
  - unpack\_sms\_SLQSGetTransLayerInfo, [1486](#)
  - unpack\_sms\_SLQSGetTransNWRRegInfo, [1487](#)
  - unpack\_sms\_SLQSModifySMSStatus, [1487](#)
  - unpack\_sms\_SLQSNWRRegInfoCallback\_ind, [1487](#)
  - unpack\_sms\_SLQSSendAsyncSMS, [1488](#)
  - unpack\_sms\_SLQSSetIndicationRegister, [1488](#)
  - unpack\_sms\_SLQSSetSmsBroadcastActivation, [1488](#)
  - unpack\_sms\_SLQSSetSmsBroadcastConfig, [1489](#)
  - unpack\_sms\_SLQSSetSmsStorage, [1489](#)
  - unpack\_sms\_SLQSSmsGetMaxStorageSize, [1489](#)
  - unpack\_sms\_SLQSSmsGetMessageProtocol, [1490](#)
  - unpack\_sms\_SLQSSmsSetRoutes, [1490](#)
  - unpack\_sms\_SLQSSwiGetSMSStorage, [1490](#)
  - unpack\_sms\_SLQSTransLayerInfoCallback\_ind, [1491](#)
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind, [1491](#)
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind, [1492](#)
  - unpack\_sms\_SLQSWmsMessageWaitingCallBack\_ind, [1492](#)
  - unpack\_sms\_SaveSMS, [1482](#)
  - unpack\_sms\_SendSMS, [1483](#)
  - unpack\_sms\_SetNewSMSCallback, [1483](#)
  - unpack\_sms\_SetNewSMSCallback\_ind, [1483](#)
  - unpack\_sms\_SetSMSCAddress, [1484](#)
- sms\_BroadcastConfig, [595](#)
  - fromServiceId, [595](#)
  - selected, [595](#)
  - toServiceId, [595](#)
- sms\_CDMABroadcastConfig, [595](#)
  - language, [596](#)
  - selected, [596](#)
  - serviceCategory, [596](#)
- sms\_getIndicationReg, [596](#)
  - pRegCallStatInfoEvt, [597](#)
  - pRegTransLayerInfoEvt, [597](#)
  - pRegTransNWRRegInfoEvt, [597](#)
- sms\_getMsgWaitingInfo, [597](#)
  - msgWaitInfo, [597](#)
  - numInstances, [597](#)
- sms\_getTransLayerInfo, [597](#)
  - pRegInd, [598](#)
  - pTransLayerInfo, [598](#)
- sms\_getTransNWRRegInfo, [598](#)
  - pRegStatus, [598](#)
- sms\_maxStorageSizeReq, [599](#)
  - pMessageMode, [599](#)
  - storageType, [599](#)
- sms\_maxStorageSizeResp, [599](#)

- freeSlots, 600
- maxStorageSize, 600
- sms\_messageWaitingInfoContent, 600
  - activeInd, 600
  - msgCount, 600
  - msgType, 600
- sms\_msgProtocolResp, 601
  - msgProtocol, 601
- sms\_qaQmi3GPP2BroadcastCfgInfo, 601
  - activated\_ind, 602
  - CDMABroadcastConfig, 602
  - num\_instances, 602
- sms\_qaQmi3GPPBroadcastCfgInfo, 602
  - activated\_ind, 602
  - broadcastConfig, 602
  - num\_instances, 602
- sms\_routeEntry, 602
  - messageClass, 603
  - messageType, 603
  - receiptAction, 603
  - routeStorage, 603
- sms\_sendAsynsmsgParams, 603
  - messageFormat, 605
  - messageSize, 605
  - pFollowOnDC, 605
  - pForceOnDC, 605
  - pLinktimer, 605
  - pMessage, 605
  - pRetryMessage, 605
  - pRetryMessageId, 605
  - pServiceOption, 605
  - pSmsOnIms, 605
  - pUserData, 605
- sms\_setIndicationReg, 606
  - pRegCallStatInfoEvt, 606
  - pRegTransLayerInfoEvt, 606
  - pRegTransNWRegInfoEvt, 606
- sms\_setRoutesReq, 606
  - numOfRoutes, 607
  - pTransferStatusReport, 607
  - routeList, 607
- sms\_transLayerInfo, 607
  - TransCap, 607
  - TransType, 608
- smsFormat
  - ims\_SMSFmtInfo, 94
- smsOnIMS
  - sMSOnIMSInfo, 611
- SmsRat
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- SmsRatVal
  - imsa\_SmsRatInfo, 100
- SmsService
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- SmsServiceRat
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- SmsServiceStatus
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- smsStorage
  - pack\_sms\_SLQSSetSmsStorage\_t, 496
- SmsSvcStatus
  - imsa\_SmsSvcStatusInfo, 101
- smsolPNW
  - ims\_SMSolPNwInfo, 95
- snr
  - loc\_satelliteInfo, 165
  - lteSSInfo, 170
  - nas\_SccRxInfo, 317
- snrlevel
  - nas\_lteSnrinformation, 248
- soMask
  - qmiWSDDataBearerTechnology, 587
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, 1093
- Source
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 975
- source
  - altSrcInfo\_t, 39
  - unpack\_dms\_GetNetworkTime\_t, 685
  - unpack\_dms\_GetNetworkTimeV2\_t, 686
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_ind\_t, 704
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 705
  - wds\_sourceOfChange, 1175
- sourceIPMask
  - LibPackTFTIDParams, 144
- SourceLength
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 975
- SourceRAT
  - imsa\_RatHandoverStatusInfo, 100
- spc
  - pack\_dms\_ResetToFactoryDefaults\_t, 382
  - pack\_nas\_SetACCOLC\_t, 428
  - pack\_wds\_SetMobileIPProfile\_t, 566
- spn
  - nas\_serviceProviderName, 317
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- spnEncoding
  - unpack\_nas\_SLQSGetPLMNName\_t, 821
- spnLength
  - nas\_serviceProviderName, 317
  - unpack\_nas\_SLQSGetPLMNName\_t, 822
- squal
  - nas\_UMTSExtInfo, 337
- srcPortRangeEnd
  - LibPackTFTIDParams, 144
- srcPortRangeStart
  - LibPackTFTIDParams, 144
- srcTiv
  - unpack\_wds\_SLQSSwiProfileChangeCallback\_ind\_t, 1096
- srvCapValid

- nas\_NR5GSystemInfoTlv, 276
- srvCapability
  - nas\_detailSvcInfo, 196
  - nas\_NR5GSysInfoTlv, 272
  - nas\_sysInfoCommon, 329
- srvCapabilityValid
  - nas\_NR5GSysInfoTlv, 272
  - nas\_sysInfoCommon, 329
- srvDomain
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 276
  - nas\_sysInfoCommon, 329
- SrvDomainPref
  - nas\_SrvDomainPrefTlv, 325
  - NASServDomainPrefTlv, 370
- srvDomainValid
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 276
  - nas\_sysInfoCommon, 329
- srvOption
  - voice\_arrSvcOption, 1119
- SrvRegRestriction
  - nas\_SrvRegRestrictionTlv, 325
  - nas\_SrvRegRestrictTlv, 326
- srvStatus
  - nas\_detailSvcInfo, 196
  - nas\_GSMSrvStatusInfo, 210
  - nas\_NR5GSerStatTlv, 267
  - nas\_NR5GSrvStatusTlv, 268
  - nas\_SrvStatusInfo, 327
- srvcapability
  - nas\_NR5GSystemInfoTlv, 276
- srxLvl
  - nas\_UMTSExtInfo, 337
- srxlev
  - nas\_cellParams, 186
  - nas\_gsmCellInfo, 209
  - nas\_umtsLTENbrCell, 342
  - nas\_wcdmaCellInfo, 345
- stage
  - uim\_refreshevent, 644
- State
  - nas\_NetworkStat1x, 262
  - nas\_NetworkStatEVDO, 263
- state
  - pack\_loc\_SLQSLOCSetCradleMountConfig\_t, 421
  - unpack\_dms\_GetActivationState\_t, 662
  - unpack\_dms\_SLQSUIMGetState\_t, 728
  - unpack\_omaDmConfigTlv\_t, 889
  - unpack\_omaDmFotaTlv\_t, 891
  - unpack\_qos\_QosFlowInfoState\_t, 911
- statmask
  - pack\_wds\_GetPacketStatus\_t, 559
- StatsMask
  - wds\_transferStatInd, 1176
- statsMask
  - wds\_TrStatInd, 1177
- StatsPeriod
  - wds\_transferStatInd, 1176
- statsPeriod
  - wds\_TrStatInd, 1177
- Status
  - unpack\_loc\_GetOpMode\_Ind\_t, 774
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 975
- status
  - loc\_FixCriteriaStatusTlv, 154
  - nas\_lteEARFCN, 227
  - nas\_ltePCI, 243
  - nas\_wcdmaUARFCN, 352
  - pack\_sms\_SetNewSMSCallback\_t, 490
  - unpack\_loc\_BestAvailPos\_Ind\_t, 767
  - unpack\_loc\_DeleteAssistData\_Ind\_t, 769
  - unpack\_loc\_InjectPositionCallback\_Ind\_t, 777
  - unpack\_loc\_InjectUTCTimeCallback\_Ind\_t, 780
  - unpack\_loc\_SetExtPowerConfig\_Ind\_t, 788
  - unpack\_loc\_SetOperationMode\_Ind\_t, 790
  - unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, 917
  - unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, 919
  - unpack\_swima\_SLQSOMADMGetSessionInfo-Ext\_t, 978
- storageIndex
  - FMSImageldElement, 80
- storageType
  - pack\_sms\_SaveSMS\_t, 489
  - pack\_sms\_SLQSDeleteSMS\_t, 491
  - pack\_sms\_SLQSGetSMS\_t, 492
  - pack\_sms\_SLQSGetSMSList\_t, 493
  - pack\_sms\_SLQSModifySMSStatus\_t, 494
  - sms\_maxStorageSizeReq, 599
  - sMSMTMessageInfo, 611
  - unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, 950
- Streaming Download Protocol (lte-fw), 32
- String
  - unpack\_dms\_GetDeviceHardwareRev\_t, 674
  - unpack\_dms\_GetDeviceMfr\_t, 675
  - unpack\_dms\_GetFSN\_t, 681
  - unpack\_dms\_UIMGetICCID\_t, 731
- stringSize
  - unpack\_dms\_GetDeviceHardwareRev\_t, 674
  - unpack\_dms\_GetDeviceMfr\_t, 675
  - unpack\_dms\_UIMGetICCID\_t, 731
- subAddr
  - voice\_calledPartySubAdd, 1123
- subAddrLen
  - voice\_calledPartySubAdd, 1123
- subAddrType
  - voice\_calledPartySubAdd, 1123
- subnetMask
  - unpack\_qos\_IPv4Addr\_t, 907
- SubnetMaskV4
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- SubsCapList

- dms\_devCurSubsCaps, [58](#)
- SubsCfgList
  - dms\_devMultiSimCaps, [60](#)
- SubsCfgListLen
  - dms\_devMultiSimCaps, [60](#)
- SubsDevList
  - dms\_devMaxCfgListCaps, [59](#)
- SubsFeatureLen
  - dms\_devSubsFeatureModeCaps, [63](#)
- SubsFeatureList
  - dms\_devSubsFeatureModeCaps, [63](#)
- SubsList
  - dms\_devSubsCfgList, [62](#)
  - dms\_devSubsList, [63](#)
- SubsListLen
  - dms\_devSubsCfgList, [62](#)
  - dms\_devSubsList, [63](#)
- subType
  - pack\_voice\_SLQSVoiceBindSubscription\_t, [539](#)
- SubsVoiceDataCap
  - dms\_devSubsVoiceDataList, [64](#)
- SubsVoiceDataCapLen
  - dms\_devSubsVoiceDataCaps, [64](#)
- SubsVoiceDataList
  - dms\_devSubsVoiceDataCaps, [64](#)
- subscrTmr
  - ims\_SubscrTmrInfo, [95](#)
- SupUSBComps
  - unpack\_dms\_GetUSBComp\_t, [692](#)
- SuppOA
  - voice\_CUGInfo, [1137](#)
- SuppPrefCUG
  - voice\_CUGInfo, [1137](#)
- supportedLteBandLen
  - dms\_LteBandsSupport, [65](#)
- svInfoMask
  - loc\_satelliteInfo, [165](#)
- svListLen
  - loc\_satelliteInfo, [166](#)
- svStatus
  - loc\_satelliteInfo, [166](#)
- svc
  - pack\_qmi\_t, [485](#)
- SvcClass
  - voice\_callFWExtInfo, [1127](#)
  - voice\_callFWInfo, [1128](#)
- SvcStatus
  - voice\_callFWExtInfo, [1127](#)
  - voice\_callFWInfo, [1128](#)
- svcType
  - voice\_ccSUPSType, [1132](#)
  - voice\_SUPSInfo, [1155](#)
- sw1
  - uim\_cardResult, [627](#)
  - uim\_cardResultInfo, [627](#)
- sw2
  - uim\_cardResult, [627](#)
  - uim\_cardResultInfo, [628](#)
- swVerString
  - pack\_dms\_SLQSSwiSetHostDevInfo\_t, [392](#)
  - unpack\_dms\_SLQSSwiGetHostDevInfo\_t, [724](#)
- swi\_uint256\_clear\_bit
  - switype\_256bit.h, [1544](#)
- swi\_uint256\_get\_bit
  - switype\_256bit.h, [1544](#)
- swi\_uint256\_print\_mask
  - switype\_256bit.h, [1544](#)
- swi\_uint256\_set\_bit
  - switype\_256bit.h, [1544](#)
- swi\_uint256\_t, [613](#)
  - word, [613](#)
- SwiDataTypes.h, [1516](#)
  - BOOL, [1517](#)
  - BYTE, [1517](#)
  - CHAR, [1517](#)
  - FLOAT, [1517](#)
  - INT32, [1517](#)
  - INT8, [1517](#)
  - LPCSTR, [1517](#)
  - qmuint16, [1517](#)
  - qmulong, [1517](#)
  - SHORT, [1517](#)
  - SWI\_API, [1517](#)
  - ULONG, [1517](#)
  - ULONGLONG, [1517](#)
  - UNUSEDPARAM, [1517](#)
  - USHORT, [1517](#)
  - WORD, [1517](#)
- swiaudio.h, [1492](#)
  - pack\_swiaudio\_SLQSGetM2MAVMute, [1495](#)
  - pack\_swiaudio\_SLQSGetM2MAudioProfile, [1494](#)
  - pack\_swiaudio\_SLQSGetM2MAudioVolume, [1494](#)
  - pack\_swiaudio\_SLQSGetM2MSprkGain, [1495](#)
  - pack\_swiaudio\_SLQSSetM2MAVMute, [1497](#)
  - pack\_swiaudio\_SLQSSetM2MAudioAVCFG, [1496](#)
  - pack\_swiaudio\_SLQSSetM2MAudioLPBK, [1496](#)
  - pack\_swiaudio\_SLQSSetM2MAudioNVDef, [1496](#)
  - pack\_swiaudio\_SLQSSetM2MAudioProfile, [1497](#)
  - pack\_swiaudio\_SLQSSetM2MAudioVolume, [1497](#)
  - pack\_swiaudio\_SLQSSetM2MSprkGain, [1498](#)
  - unpack\_swiaudio\_SLQSGetM2MAVMute, [1499](#)
  - unpack\_swiaudio\_SLQSGetM2MAudioProfile, [1498](#)
  - unpack\_swiaudio\_SLQSGetM2MAudioVolume, [1499](#)
  - unpack\_swiaudio\_SLQSGetM2MSprkGain, [1499](#)
  - unpack\_swiaudio\_SLQSSetM2MAVMute, [1501](#)
  - unpack\_swiaudio\_SLQSSetM2MAVMute\_t, [1494](#)
  - unpack\_swiaudio\_SLQSSetM2MAudioAVCFG, [1500](#)
  - unpack\_swiaudio\_SLQSSetM2MAudioAVCFG\_t, [1494](#)
  - unpack\_swiaudio\_SLQSSetM2MAudioLPBK, [1500](#)
  - unpack\_swiaudio\_SLQSSetM2MAudioLPBK\_t, [1494](#)



- unpack\_swiaudio\_SLQSSetM2MAudioNVDef, 1500
- unpack\_swiaudio\_SLQSSetM2MAudioNVDef\_t, 1494
- unpack\_swiaudio\_SLQSSetM2MAudioProfile, 1501
- unpack\_swiaudio\_SLQSSetM2MAudioProfile\_t, 1494
- unpack\_swiaudio\_SLQSSetM2MAudioVolume, 1501
- unpack\_swiaudio\_SLQSSetM2MAudioVolume\_t, 1494
- unpack\_swiaudio\_SLQSSetM2MSpkrGain, 1502
- unpack\_swiaudio\_SLQSSetM2MSpkrGain\_t, 1494
- swiaudio\_PCMparams, 613
- iFaceTab, 614
- iFaceTabLen, 614
- swiavms.h, 1502
  - pack\_swiavms\_SLQSAVMSSetSettings, 1504
  - pack\_swiavms\_SLQSAVMSSetSettings\_v2, 1505
  - pack\_swiavms\_SLQSAVMSSendSelection, 1506
  - pack\_swiavms\_SLQSAVMSSessionGetInfo, 1506
  - pack\_swiavms\_SLQSAVMSSetSettings, 1507
  - pack\_swiavms\_SLQSAVMSSetSettings\_v2, 1508
  - pack\_swiavms\_SLQSAVMSSetSettingsNoAuto-RebootField, 1509
  - pack\_swiavms\_SLQSAVMSStartSession, 1509
  - pack\_swiavms\_SLQSAVMSStopSession, 1510
  - pack\_swiavms\_SLQSAvmsSetEventReport, 1507
  - unpack\_swiavms\_SLQSAVMSEventReportInd, 1511
  - unpack\_swiavms\_SLQSAVMSSetSettings, 1511
  - unpack\_swiavms\_SLQSAVMSSetSettings\_v2, 1511
  - unpack\_swiavms\_SLQSAVMSSendSelection, 1512
  - unpack\_swiavms\_SLQSAVMSSessionGetInfo, 1512
  - unpack\_swiavms\_SLQSAVMSSetSettings, 1513
  - unpack\_swiavms\_SLQSAVMSSetSettings\_v2, 1514
  - unpack\_swiavms\_SLQSAVMSStartSession, 1514
  - unpack\_swiavms\_SLQSAVMSStopSession, 1515
  - unpack\_swiavms\_SLQSAVMSStopSession\_avc2, 1515
  - unpack\_swiavms\_SLQSAVMSStopSession\_avc2\_t, 1504
  - unpack\_swiavms\_SLQSAvmsSetEventReport, 1513
- swidms.h, 1517
  - pack\_swidms\_SLQSSwiDmsGetHWWatchdog, 1519
  - pack\_swidms\_SLQSSwiDmsGetMTU, 1519
  - pack\_swidms\_SLQSSwiDmsGetSecureInfo, 1519
  - pack\_swidms\_SLQSSwiDmsGetUsbComp, 1520
  - pack\_swidms\_SLQSSwiDmsGetUsbNetNum, 1520
  - pack\_swidms\_SLQSSwiDmsSetHWWatchdog, 1520
  - pack\_swidms\_SLQSSwiDmsSetMTU, 1521
  - pack\_swidms\_SLQSSwiDmsSetUsbComp, 1521
  - pack\_swidms\_SLQSSwiDmsSetUsbNetNum, 1521
  - unpack\_swidms\_SLQSSwiDmsGetHWWatchdog, 1522
  - unpack\_swidms\_SLQSSwiDmsGetMTU, 1522
  - unpack\_swidms\_SLQSSwiDmsGetSecureInfo, 1522
  - unpack\_swidms\_SLQSSwiDmsGetUsbComp, 1523
  - unpack\_swidms\_SLQSSwiDmsGetUsbNetNum, 1523
  - unpack\_swidms\_SLQSSwiDmsSetHWWatchdog, 1524
  - unpack\_swidms\_SLQSSwiDmsSetMTU, 1524
  - unpack\_swidms\_SLQSSwiDmsSetUsbComp, 1524
  - unpack\_swidms\_SLQSSwiDmsSetUsbNetNum, 1525
  - unpack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t, 1519
- swidms\_SwiDmsGetHWWatchdog, 617
  - count, 618
  - enable, 618
  - resetDelay, 618
  - timeout, 618
- swidms\_ehrpdMTUSizeTlv, 614
  - ehrpMTUSize, 614
  - TlvPresent, 614
- swidms\_hrpMTUSizeTlv, 614
  - hrpMTUSize, 615
  - TlvPresent, 615
- swidms\_interfaceCfgTlv, 615
  - CfgValue, 616
  - CurrentCfgType, 616
  - TlvPresent, 616
- swidms\_mtuSize3gppTlv, 616
  - MTUSize3gpp, 617
  - TlvPresent, 617
- swidms\_supportedIntBitmaskTlv, 617
  - TlvPresent, 617
  - ValidBitmasks, 617
- swidms\_usbMTUSizeTlv, 618
  - TlvPresent, 618
  - UsbMTUSize, 618
- swiloc.h, 1525
  - pack\_swiloc\_SwiLocGetAutoStart, 1526
  - pack\_swiloc\_SwiLocSetAutoStart, 1526
  - unpack\_swiloc\_SwiLocGetAutoStart, 1526
  - unpack\_swiloc\_SwiLocSetAutoStart, 1527
  - unpack\_swiloc\_SwiLocSetAutoStart\_t, 1525
- swioma.h, 1527
  - pack\_swioma\_SLQSOMADMAAlertCallback, 1528
  - pack\_swioma\_SLQSOMADMCancelSession, 1529

- pack\_swioama\_SLQSOMADMGetSessionInfo, 1530
- pack\_swioama\_SLQSOMADMGetSettings, 1530
- pack\_swioama\_SLQSOMADMGetSelection, 1531
- pack\_swioama\_SLQSOMADMSetSettings, 1531
- pack\_swioama\_SLQSOMADMStartSession, 1532
- unpack\_swioama\_SLQSOMADMAAlertCallback, 1533
- unpack\_swioama\_SLQSOMADMAAlertCallback\_ind, 1533
- unpack\_swioama\_SLQSOMADMAAlertCallback\_t, 1528
- unpack\_swioama\_SLQSOMADMCancelSession, 1534
- unpack\_swioama\_SLQSOMADMCancelSession\_t, 1528
- unpack\_swioama\_SLQSOMADMGetSessionInfo, 1534
- unpack\_swioama\_SLQSOMADMGetSettings, 1535
- unpack\_swioama\_SLQSOMADMGetSelection, 1535
- unpack\_swioama\_SLQSOMADMGetSelection\_t, 1528
- unpack\_swioama\_SLQSOMADMSetSettings, 1536
- unpack\_swioama\_SLQSOMADMSetSettings\_t, 1528
- unpack\_swioama\_SLQSOMADMStartSession, 1536
- swioamaext.h, 1537
  - pack\_swioama\_SLQSOMADMCancelSessionExt, 1538
  - pack\_swioama\_SLQSOMADMGetSessionInfoExt, 1538
  - pack\_swioama\_SLQSOMADMGetSelectionExt, 1539
  - pack\_swioama\_SLQSOMADMSetSettingsExt, 1539
  - pack\_swioama\_SLQSOMADMStartSessionExt, 1540
  - unpack\_swioama\_SLQSOMADMCancelSession-Ext, 1541
  - unpack\_swioama\_SLQSOMADMCancelSession-Ext\_t, 1537
  - unpack\_swioama\_SLQSOMADMGetSessionInfo-Ext, 1541
  - unpack\_swioama\_SLQSOMADMGetSelectionExt, 1542
  - unpack\_swioama\_SLQSOMADMGetSelectionExt\_t, 1538
  - unpack\_swioama\_SLQSOMADMSetSettingsExt, 1542
  - unpack\_swioama\_SLQSOMADMSetSettingsExt\_t, 1538
  - unpack\_swioama\_SLQSOMADMStartSessionExt, 1543
  - unpack\_swioama\_SLQSOMADMStartSessionExt\_t, 1538
- switchOption
  - pack\_voice\_SLQSVoiceALSSetLineSwitching\_t, 538
  - switype\_256bit.h, 1543
    - logger, 1544
    - swi\_uint256\_clear\_bit, 1544
    - swi\_uint256\_get\_bit, 1544
    - swi\_uint256\_print\_mask, 1544
    - swi\_uint256\_set\_bit, 1544
  - sysForbidden
    - nas\_NR5GSystemInfoTlv, 276
  - sysForbiddenValid
    - nas\_NR5GSystemInfoTlv, 276
  - sysInfoCDMA
    - nas\_CDMA SysInfo, 185
  - sysInfoGSM
    - nas\_GSM SysInfo, 213
  - sysInfoHDR
    - nas\_HDR SysInfo, 219
  - sysInfoLTE
    - nas\_LTE SysInfo, 252
  - sysInfoWCDMA
    - nas\_WCDMA SysInfo, 351
  - system
    - loc\_satelliteInfo, 166
    - loc\_SV, 167
  - systemDiscontinuities
    - pack\_pds\_PDSInjectTimeReference\_t, 474
  - SystemID
    - unpack\_nas\_SLQSGetServingSystem\_t, 826
    - unpack\_nas\_SLQSGetServingSystemV2\_t, 831
  - systemID
    - nas\_CDMA SysInfo, 185
  - systemMode
    - nas\_CommInfo, 190
  - systemTime
    - pack\_pds\_PDSInjectTimeReference\_t, 474
  - szAPN
    - PackSwiAvmsSetSettingsAPNInfo, 583
  - szAlertMsg
    - UnpackSwiAvmsEventReportConfig, 1103
  - szCarrier\_str
    - \_litew\_FirmwareInfo, 37
  - szCarrierPriversion\_str
    - \_litew\_FirmwareInfo, 37
  - szDescription
    - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1101
  - szFwversion\_str
    - \_litew\_FirmwareInfo, 37
  - szModelid\_str
    - \_litew\_FirmwareInfo, 37
  - szName
    - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1101
  - szPWD
    - PackSwiAvmsSetSettingsAPNInfo, 583
  - szPackageid\_str
    - \_litew\_FirmwareInfo, 37

- szPartno\_str
  - \_litefw\_FirmwarePartNo\_, 38
- szSku\_str
  - \_litefw\_FirmwareInfo\_, 37
- szUname
  - PackSwiAvmsSetSettingsAPNInfo, 583
- szVersion
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- t3396\_apn
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 875
- t3396\_plmn\_id
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 876
- t3396\_val
  - unpack\_nas\_SLQSNasTimerCallback\_ind\_t, 876
- TCPDstPort
  - unpack\_qos\_swiQosFilter\_t, 922
- TCPSrcPort
  - unpack\_qos\_swiQosFilter\_t, 922
- TDSCDMAECIOThreshListLen
  - nas\_TDSCDMAECIOThresh, 330
- TDSCDMARSCPTThreshListLen
  - nas\_TDSCDMARSCPTThresh, 331
- TDSCDMARSSIThreshListLen
  - nas\_TDSCDMARSSIThresh, 331
- TDSCDMASINRThreshListLen
  - nas\_TDSCDMASINRThresh, 332
- TLVPresent
  - dms\_LteBandsSupport, 65
  - nas\_ImVoiceSupportLteTlv, 221
  - nas\_LteEmbmsCoverageTlv, 228
  - nas\_LteEmbmsTraceIdTlv, 228
  - nas\_LteRegDomainTlv, 244
  - nas\_LteVoiceDomainTlv, 253
  - nas\_MNCPCSDigitStatus, 256
  - nas\_NR5GCellStatusInfoTlv, 266
  - nas\_NR5GSrvStatusTlv, 269
  - nas\_NR5GSysInfoTlv, 272
  - nas\_SimRejInfoTlv, 321
  - nas\_SrvRegRestrictionTlv, 325
- TMD\_MAX\_DEV\_LIST
  - tmd.h, 1545
- TMTlv
  - unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t, 745
- TPCause
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack-\_ind\_t, 949
- TRMessageTlv
  - unpack\_sms\_SetNewSMSCallback\_ind\_t, 935
- TT1Tlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, 749
- TT2Tlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, 749
- TTfTlv
  - unpack\_ims\_SLQSSIPCfgCallBack\_ind\_t, 749
- TX\_PWR
  - nas\_NetworkStat1x, 262
- TXBOTlv
  - unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, 1072
- TXChan
  - nas\_LTEInfo, 232
- tXDroppedCount
  - unpack\_wds\_GetPacketStatus\_t, 1066
- tXOKBytesLastCall
  - unpack\_wds\_GetPacketStatus\_t, 1066
- tXOkBytesCount
  - unpack\_wds\_GetPacketStatus\_t, 1066
- tXPacketErrors
  - unpack\_wds\_GetPacketStatus\_t, 1066
- tXPacketOverflows
  - unpack\_wds\_GetPacketStatus\_t, 1066
- tXPacketSuccesses
  - unpack\_wds\_GetPacketStatus\_t, 1066
- Tables, 33
- tac
  - nas\_LTEInfoIntrafreq, 234
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 276
- tacValid
  - nas\_LTESysInfo, 252
  - nas\_NR5GSysInfoTlv, 272
  - nas\_NR5GSystemInfoTlv, 276
- TargetRAT
  - imsa\_RatHandoverStatusInfo, 100
- TdsBandCapability
  - unpack\_dms\_SLQSGetBandCapability\_t, 711
  - unpack\_dms\_SLQSGetBandCapabilityExt\_t, 715
- TdscdmaBandPref
  - nas\_TDSCDMABandPrefTlv, 330
- tdscdmaSigInfoExt, 619
  - ecio, 619
  - rscp, 619
  - rss, 619
  - sinr, 619
- techName
  - unpack\_wds\_SLQSSetPacketSrvStatusCallBack-\_t, 1087
- Technology
  - nas\_DeviceConfigDetail, 197
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- TechnologyPref
  - pack\_nas\_SetNetworkPreference\_t, 430
- tempData\_t, 619
  - temperature, 620
  - temperatureDataLen, 620
  - timeOfFirstSample, 620
  - timeOffset, 620
  - timeSource, 620
- TempStat
  - dms\_TemperatureTlv, 72
- TempTlv
  - unpack\_dms\_SwiEventReportCallBack\_ind\_t, 729
- Temperature
  - dms\_TemperatureTlv, 72



- temperature
  - nas\_CommInfo, [190](#)
  - tempData\_t, [620](#)
- temperatureDataLen
  - tempData\_t, [620](#)
- threshGsmHigh
  - nas\_lteGsmCellInfo, [230](#)
- threshGsmLow
  - nas\_lteGsmCellInfo, [230](#)
- threshServingLow
  - nas\_LTEInfoIntraFreq, [234](#)
- threshXHigh
  - nas\_infoInterFreq, [222](#)
- threshXLow
  - nas\_infoInterFreq, [223](#)
- threshXhigh
  - nas\_lteWcdmaCellInfo, [253](#)
- threshXlow
  - nas\_lteWcdmaCellInfo, [254](#)
- ThrottleTimer
  - LibPackPDNThrottleTimer, [106](#)
- Time
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, [975](#)
- time
  - NASTimeInfoTlv, [372](#)
- Time\_uncert\_ms
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- timeBase
  - pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, [479](#)
- TimeLength
  - unpack\_swioma\_SLQSOMADMGetSessionInfo\_t, [975](#)
- timeMsec
  - pack\_loc\_SLQSLOCInjectUTCTime\_t, [420](#)
  - pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, [479](#)
- timeOfFirstSample
  - sensorData\_t, [592](#)
  - tempData\_t, [620](#)
- timeOffset
  - sensorData\_t, [592](#)
  - tempData\_t, [620](#)
- timeSource
  - tempData\_t, [620](#)
- TimeStmp\_gps\_week
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- TimeStmp\_tow\_ms
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- timeSyncRefCounter
  - unpack\_loc\_EventTimeSyncCallback\_Ind\_t, [771](#)
- timeTlv
  - NASQmiCbkNasSwiOTAMessageInd, [366](#)
- timeUncMsec
  - pack\_loc\_SLQSLOCInjectUTCTime\_t, [420](#)
  - pack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t, [480](#)
- timeZone
  - nas\_timeInfo, [334](#)
- timeout
  - pack\_pds\_SetPDSDefaults\_t, [476](#)
  - pack\_qmi\_t, [485](#)
  - pack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, [509](#)
  - swidms\_SwiDmsGetHWWatchdog, [618](#)
- Timers
  - PackSwiAvmsSetSettingsConnectionRetryTimers, [584](#)
  - PackSwiAVMSSettingsConnectionRetryTimers, [586](#)
- timestamp
  - unpack\_dms\_GetNetworkTime\_t, [685](#)
  - unpack\_dms\_GetNetworkTimeV2\_t, [686](#)
- timestampAge
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- timestampUtc
  - pack\_loc\_SLQSLOCInjectPosition\_t, [418](#)
- timingAdvance
  - nas\_GERANInfo, [206](#)
- TlvPresenceMask
  - qmTlvResult, [588](#)
- TlvPresent
  - cat\_commonEventTlv, [50](#)
  - dms\_ActivationStatusTlv, [56](#)
  - dms\_OperatingModeTlv, [66](#)
  - dms\_PSMActiveTimerIndTlv, [66](#)
  - dms\_PSMActiveTimerTlv, [67](#)
  - dms\_PSMDurationDueToOOSTlv, [67](#)
  - dms\_PSMDurationThresholdTlv, [68](#)
  - dms\_PSMEarlyWakeupTimeTlv, [68](#)
  - dms\_PSMEnableStateIndTlv, [69](#)
  - dms\_PSMEnableStateTlv, [70](#)
  - dms\_PSMPeriodicUpdateTimerIndTlv, [70](#)
  - dms\_PSMPeriodicUpdateTimerTlv, [71](#)
  - dms\_PSMRandomizationWindowTlv, [71](#)
  - dms\_TemperatureTlv, [72](#)
  - dms\_UimAutoSwitchActSlotTlv, [72](#)
  - dms\_UimStatusTlv, [73](#)
  - dms\_VoltageTlv, [74](#)
  - eTWSPLMNInfoTlv, [78](#)
  - ims\_AMRModelInfo, [85](#)
  - ims\_AMROctAlgnInfo, [85](#)
  - ims\_AMRWBModelInfo, [86](#)
  - ims\_AMRWBOctAlgnInfo, [86](#)
  - ims\_CSCFPortNameInfo, [87](#)
  - ims\_EnabAMRWBInfo, [87](#)
  - ims\_EnabSCRAMRInfo, [88](#)
  - ims\_EnabSCRAMRWBInfo, [88](#)
  - ims\_IMSDomainInfo, [89](#)
  - ims\_IMSTestModelInfo, [89](#)
  - ims\_MinSessExplInfo, [90](#)
  - ims\_PCSCFPortInfo, [90](#)
  - ims\_PhCtxtURIInfo, [91](#)
  - ims\_RngBkTmrInfo, [91](#)
  - ims\_RngTmrInfo, [92](#)

- ims\_RTPRTCPInactTmrDurInfo, 92
- ims\_SessDurInfo, 93
- ims\_SigCompEnInfo, 93
- ims\_SIPPortInfo, 93
- ims\_SIPRegnTmrInfo, 94
- ims\_SMSFmtInfo, 94
- ims\_SMSoIPNwInfo, 95
- ims\_SubscrTmrInfo, 95
- ims\_TmrT1Info, 96
- ims\_TmrT2Info, 96
- ims\_TmrTfInfo, 97
- imsa\_IMSFailErrCodeTlv, 97
- imsa\_IMSRegStatusErrorInfo, 98
- imsa\_IMSRegStatusInfo, 98
- imsa\_NewIMSRegStatusInfo, 99
- imsa\_RatHandoverStatusInfo, 100
- imsa\_SmsRatInfo, 101
- imsa\_SmsSvcStatusInfo, 101
- imsa\_UtRatInfo, 102
- imsa\_UtSvcStatusInfo, 102
- imsa\_VoipRatInfo, 103
- imsa\_VoipSvcStatusInfo, 103
- imsa\_VtRatInfo, 104
- imsa\_VtSvcStatusInfo, 104
- loc\_AppProviderInfoTlv, 150
- loc\_FixCriteriaStatusTlv, 154
- loc\_HorAccuracyLvITlv, 158
- loc\_IntermediateRptStateTlv, 159
- loc\_IPv4Info, 160
- loc\_IPv6Info, 161
- loc\_MinIntervalTlv, 163
- loc\_urlAddr, 169
- messageModeTlv, 171
- nas\_AcqOrderPrefTlv, 172
- nas\_BandPrefInfoTlv, 175
- nas\_BandPrefTlv, 177
- nas\_CiotAcqOrderPrefTlv, 188
- nas\_CiotLteOpModePrefTlv, 188
- nas\_CsgId, 190
- nas\_EdrxCiotLteMode, 200
- nas\_EdrxCycleLength, 200
- nas\_EdrxEnableType, 201
- nas\_EdrxPagingTimeWindow, 201
- nas\_EdrxRatType, 202
- nas\_EmerModeTlv, 203
- nas\_ForbiddenNetworks3GPP, 205
- nas\_GWAcqOrderPrefTlv, 214
- nas\_homeNwMNC3GppTlv, 220
- nas\_LTEBandPrefTlv, 226
- nas\_LteCiotOpModeTlv, 226
- nas\_LteEarfcnInfo, 227
- nas\_LteM1BandPrefTlv, 238
- nas\_LteNb1BandPrefTlv, 240
- nas\_LTEOperationalModeTlv, 240
- nas\_LTEOperationMode, 241
- nas\_LteOpMode, 241
- nas\_LteOpModeTlv, 242
- nas\_ModePrefTlv, 257
- nas\_NetSelPrefTlv, 259
- nas\_networkNameSrcTlv, 260
- nas\_NR5GCellStatusTlv, 266
- nas\_NR5GSerStatTlv, 267
- nas\_NR5GSystemInfoTlv, 276
- nas\_NumScellsConfig, 276
- nas\_nwNameSrc3GppTlv, 277
- nas\_PhyCaAggDIBW, 279
- nas\_PhyCaAggPcellInfo, 280
- nas\_PhyCaAggScellDIBw, 281
- nas\_PhyCaAggScellIndex, 281
- nas\_PhyCaAggScellIndType, 282
- nas\_PhyCaAggScellInfo, 284
- nas\_PlmnID, 286
- nas\_PRLPrefTlv, 289
- nas\_RankIndicatorTlv, 296
- nas\_RatDisabledMaskTlv, 297
- nas\_RejectReasonTlv, 297
- nas\_RfBandInfoExtFormat, 299
- nas\_RfBandInfoExtTlv, 301
- nas\_RfBandInfoTlv, 302
- nas\_RfBandwidthInfo, 303
- nas\_RfBandwidthInfoTlv, 305
- nas\_RfDedicatedBandInfo, 306
- nas\_RfDedicatedBandInfoTlv, 308
- nas\_RfInfoTlv, 309
- nas\_RoamPrefTlv, 310
- nas\_SccRxInfo, 317
- nas\_SignalStrengthTlv, 320
- nas\_SLQSSignalStrengthsTlv, 324
- nas\_SrvDomainPrefTlv, 325
- nas\_SrvRegRestrictTlv, 326
- nas\_TDSCDMABandPrefTlv, 330
- nas\_timeInfo, 334
- nas\_UMTSExtInfo, 337
- nas\_UsageSettingTlv, 344
- nas\_VoiceDomainPrefTlv, 344
- nas\_WCDMACellInfoExt, 346
- NASAcqOrderPrefTlv, 352
- NASBandPreferenceTlv, 353
- NASCiotAcqOrderPrefTlv, 353
- NASCiotLteOpModePrefTlv, 354
- NASEmergencyModeTlv, 354
- NASGWAcqOrderPrefTlv, 356
- NASLTEBandPreferenceTlv, 356
- NASLteM1BandPrefTlv, 357
- NASLteNasReleaseInfoTlv, 357
- NASLteNB1BandPrefTlv, 358
- NASModePreferenceTlv, 358
- NASNetSelPreferenceTlv, 358
- NASNr5gBandPrefTlv, 359
- NASOTAMessageTlv, 360
- NASPhyCaAggPcellInfo, 361
- NASPhyCaAggScellArray, 362
- NASPhyCaAggScellDIBw, 363
- NASPhyCaAggScellIndex, 363
- NASPhyCaAggScellIndType, 364
- NASPhyCaAggScellInfo, 365

- NASPRPreferenceTlv, [366](#)
- NASRatDisabledMaskTlv, [369](#)
- NASRoamPreferenceTlv, [369](#)
- NASServDomainPrefTlv, [370](#)
- NASTimeInfoTlv, [372](#)
- newMTMessageTlv, [372](#)
- sMSCAddressTlv, [609](#)
- sMSEtwsMessageTlv, [610](#)
- sMSOnIMSTlv, [612](#)
- swidms\_ehrpdMTUSizeTlv, [614](#)
- swidms\_hrpdmTUSizeTlv, [615](#)
- swidms\_intfaceCfgTlv, [616](#)
- swidms\_mtuSize3gppTlv, [617](#)
- swidms\_supportedIntBitmaskTlv, [617](#)
- swidms\_usbMTUSizeTlv, [618](#)
- transferRouteMessageTlv, [621](#)
- uim\_additionalReadResult, [622](#)
- uim\_cardResultInfo, [628](#)
- uim\_GetSlotsInfoTlv, [636](#)
- uim\_GetSlotsStatusTlv, [636](#)
- uim\_indToken, [637](#)
- uim\_readResultInfo, [641](#)
- uim\_UIMGetFDNStatus, [650](#)
- uim\_UIMGetHiddenKeyStatus, [650](#)
- uim\_UIMGetIndex, [651](#)
- unpack\_pds\_SetPdsState\_Ind\_t, [900](#)
- unpack\_uim\_SLQSUIRefreshCallback\_Ind\_t, [996](#)
- UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, [1102](#)
- UnpackSwiAvmsEventReportConfig, [1103](#)
- UnpackSwiAvmsEventReportConnectionRequest, [1103](#)
- UnpackSwiAvmsEventReportDataSessionStatus, [1104](#)
- UnpackSwiAvmsEventReportHTTPStatus, [1105](#)
- UnpackSwiAvmsEventReportNotification, [1107](#)
- UnpackSwiAvmsEventReportPackageID, [1107](#)
- UnpackSwiAvmsEventReportRegStatus, [1107](#)
- UnpackSwiAvmsEventReportSessionType, [1108](#)
- UnpackSwiAvmsEventReportWAMSParam-Change, [1109](#)
- wds\_ChannelRateTlv, [1158](#)
- wds\_ConnStatusTlv, [1159](#)
- wds\_DataBearTechTlv, [1161](#)
- wds\_DataULongLongTlv, [1162](#)
- wds\_DataULongTlv, [1162](#)
- wds\_DHCPLeaseOptTlv, [1163](#)
- wds\_DHCPLeaseStateTlv, [1163](#)
- wds\_DHCPProfileIdTlv, [1164](#)
- wds\_DormStatTlv, [1168](#)
- wds\_IPv4AdTlv, [1170](#)
- wds\_LastMdmCallEndRsnTlv, [1171](#)
- wds\_RXBytesOKTlv, [1175](#)
- wds\_TXBytesOKTlv, [1177](#)
- tlvPresent
  - voice\_USSDNotificationNetworkInfo, [1156](#)
- TlvResult
  - unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, [966](#)
- TlvResultCode
  - qmTlvResult, [588](#)
- Tlvresult
  - pack\_dms\_GetCustFeaturesV2\_t, [381](#)
  - pack\_dms\_SetCustFeaturesV2\_t, [386](#)
  - pack\_dms\_SetPower\_t, [388](#)
  - pack\_dms\_SetUSBComp\_t, [388](#)
  - pack\_dms\_UIMGetICCID\_t, [395](#)
  - pack\_fms\_GetImagesPreference\_t, [399](#)
  - pack\_fms\_GetStoredImages\_t, [399](#)
  - pack\_fms\_SetImagesPreference\_t, [400](#)
  - pack\_loc\_Delete\_Assist\_Data\_t, [409](#)
  - pack\_loc\_EventRegister\_t, [411](#)
  - pack\_loc\_SetExtPowerState\_t, [411](#)
  - pack\_loc\_SetOperationMode\_t, [412](#)
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, [412](#)
  - pack\_loc\_Start\_t, [424](#)
  - pack\_loc\_Stop\_t, [424](#)
  - pack\_nas\_SetNetworkPreference\_t, [431](#)
  - pack\_uim\_ChangePin\_t, [521](#)
  - pack\_uim\_ReadTransparent\_t, [522](#)
  - pack\_uim\_SetPinProtection\_t, [523](#)
  - pack\_uim\_UnblockPin\_t, [535](#)
  - pack\_uim\_VerifyPin\_t, [536](#)
  - unpack\_cat\_SetCATEventCallback\_t, [661](#)
  - unpack\_dms\_GetBandCapability\_t, [664](#)
  - unpack\_dms\_GetCrashAction\_t, [665](#)
  - unpack\_dms\_GetCustFeature\_t, [667](#)
  - unpack\_dms\_GetCustFeaturesV2\_t, [668](#)
  - unpack\_dms\_GetDeviceCap\_t, [669](#)
  - unpack\_dms\_GetDeviceCapabilitiesV2\_t, [673](#)
  - unpack\_dms\_GetDeviceHardwareRev\_t, [674](#)
  - unpack\_dms\_GetDeviceMfr\_t, [675](#)
  - unpack\_dms\_GetDeviceSerialNumbers\_t, [677](#)
  - unpack\_dms\_GetFirmwareInfo\_t, [678](#)
  - unpack\_dms\_GetFirmwareRevision\_t, [679](#)
  - unpack\_dms\_GetFirmwareRevisions\_t, [681](#)
  - unpack\_dms\_GetFSN\_t, [681](#)
  - unpack\_dms\_GetIMSI\_t, [683](#)
  - unpack\_dms\_GetManufacturer\_t, [683](#)
  - unpack\_dms\_GetModelID\_t, [684](#)
  - unpack\_dms\_GetNetworkTime\_t, [685](#)
  - unpack\_dms\_GetNetworkTimeV2\_t, [686](#)
  - unpack\_dms\_GetOfflineReason\_t, [687](#)
  - unpack\_dms\_GetPower\_t, [688](#)
  - unpack\_dms\_GetPRLVersion\_t, [689](#)
  - unpack\_dms\_GetUSBComp\_t, [692](#)
  - unpack\_dms\_GetVoiceNumber\_t, [693](#)
  - unpack\_dms\_PSMCfChange\_ind\_t, [694](#)
  - unpack\_dms\_ResetToFactoryDefaults\_t, [695](#)
  - unpack\_dms\_SetActivationStatusCallback\_t, [695](#)
  - unpack\_dms\_SetCustFeature\_t, [696](#)
  - unpack\_dms\_SetCustFeaturesV2\_t, [697](#)
  - unpack\_dms\_SetEventReport\_ind\_t, [698](#)
  - unpack\_dms\_SetEventReport\_t, [698](#)
  - unpack\_dms\_SetFirmwarePreference\_t, [699](#)

- unpack\_dms\_SetIndicationRegister\_t, 699
- unpack\_dms\_SetPower\_t, 700
- unpack\_dms\_SetUSBComp\_t, 700
- unpack\_dms\_SLQSDmsSwtGetResetInfo\_Ind\_t, 704
- unpack\_dms\_SLQSDmsSwtGetResetInfo\_t, 705
- unpack\_dms\_SLQSDmsSwtIndicationRegister\_t, 707
- unpack\_dms\_SLQSSwtGetERIFile\_t, 716
- unpack\_dms\_SLQSSwtSetPowerSaveModeConfig\_t, 718
- unpack\_dms\_SLQSSwtClearDyingGaspStatistics\_t, 718
- unpack\_dms\_SLQSSwtGetCrashInfo\_t, 719
- unpack\_dms\_SLQSSwtGetDyingGaspCfg\_t, 720
- unpack\_dms\_SLQSSwtGetDyingGaspStatistics\_t, 720
- unpack\_dms\_SLQSSwtGetFwUpdateStatus\_t, 723
- unpack\_dms\_SLQSSwtGetHostDevInfo\_t, 724
- unpack\_dms\_SLQSSwtGetOSInfo\_t, 725
- unpack\_dms\_SLQSSwtGetSerialNoExt\_t, 726
- unpack\_dms\_SLQSSwtSetDyingGaspCfg\_t, 726
- unpack\_dms\_SLQSSwtSetHostDevInfo\_t, 727
- unpack\_dms\_SLQSSwtSetOSInfo\_t, 727
- unpack\_dms\_SLQSUIMGetState\_t, 728
- unpack\_dms\_SwtSetEventReport\_t, 729
- unpack\_dms\_SwtUimSelect\_t, 730
- unpack\_dms\_UIMGetControlKeyStatus\_t, 731
- unpack\_dms\_UIMGetICCID\_t, 731
- unpack\_dms\_UIMGetPINStatus\_t, 733
- unpack\_dms\_UIMSetControlKeyProtection\_t, 734
- unpack\_dms\_UIMSetPINProtection\_t, 735
- unpack\_dms\_UIMUnblockControlKey\_t, 735
- unpack\_fms\_GetImagesPreference\_t, 736
- unpack\_fms\_GetStoredImages\_t, 737
- unpack\_fms\_SetImagesPreference\_t, 737
- unpack\_loc\_BestAvailPos\_Ind\_t, 767
- unpack\_loc\_Delete\_Assist\_Data\_t, 768
- unpack\_loc\_DeleteAssistData\_Ind\_t, 769
- unpack\_loc\_EngineState\_Ind\_t, 770
- unpack\_loc\_EventNMEA\_Ind\_t, 770
- unpack\_loc\_EventRegister\_t, 771
- unpack\_loc\_GetServer\_Ind\_t, 775
- unpack\_loc\_GnssSvInfo\_Ind\_t, 776
- unpack\_loc\_PositionRpt\_Ind\_t, 787
- unpack\_loc\_SetExtPowerConfig\_Ind\_t, 788
- unpack\_loc\_SetExtPowerState\_t, 789
- unpack\_loc\_SetOperationMode\_Ind\_t, 790
- unpack\_loc\_SetOperationMode\_t, 790
- unpack\_loc\_SetServer\_Ind\_t, 791
- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, 792
- unpack\_loc\_SLQSLOCGetOpMode\_t, 792
- unpack\_loc\_Start\_t, 793
- unpack\_loc\_Stop\_t, 793
- unpack\_nas\_GetNetworkPreference\_t, 801
- unpack\_nas\_SetNetworkPreference\_t, 812
- unpack\_nas\_SetServingSystemCallback\_ind\_t, 813
- unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 816
- unpack\_nas\_SLQSNasGetRFInfo\_t, 864
- unpack\_nas\_SLQSNasSwtOTAMessageCallback\_ind\_t, 875
- unpack\_nas\_SLQSSwtSysSelectionPrefCallBack\_ind\_t, 878
- unpack\_result\_t, 928
- unpack\_swidms\_SLQSSwtDmsSetHWWatchdog\_t, 968
- unpack\_swidms\_SLQSSwtDmsSetMTU\_t, 968
- unpack\_swidms\_SLQSSwtDmsSetUsbComp\_t, 969
- unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, 981
- unpack\_tmd\_SLQSTmdGetMitigationDevList\_t, 982
- unpack\_tmd\_SLQSTmdGetMitigationLvl\_t, 983
- unpack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, 984
- unpack\_uim\_ChangePin\_t, 985
- unpack\_uim\_GetCardStatus\_t, 985
- unpack\_uim\_GetCardStatusV2\_t, 986
- unpack\_uim\_ReadTransparent\_t, 987
- unpack\_uim\_SetPinProtection\_t, 988
- unpack\_uim\_UnblockPin\_t, 1000
- unpack\_uim\_UnblockPinV2\_t, 1001
- unpack\_uim\_VerifyPin\_t, 1002
- unpack\_wds\_SLQSCreateProfile\_t, 1070
- unpack\_wds\_SLQSSwtGetProfileSettings\_t, 1080
- unpack\_wds\_SLQSSwtGetProfileSettingsV2\_t, 1081
- unpack\_wds\_SLQSSwtSetIPFamilyPreference\_t, 1085
- tmd.h, 1545
  - pack\_tmd\_SLQSTmdDeRegNotMitigationLvl, 1546
  - pack\_tmd\_SLQSTmdGetMitigationDevList, 1546
  - pack\_tmd\_SLQSTmdGetMitigationLvl, 1546
  - pack\_tmd\_SLQSTmdRegNotMitigationLvl, 1547
  - TMD\_MAX\_DEV\_LIST, 1545
  - unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl, 1547
  - unpack\_tmd\_SLQSTmdGetMitigationDevList, 1547
  - unpack\_tmd\_SLQSTmdGetMitigationLvl, 1548
  - unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind, 1548
  - unpack\_tmd\_SLQSTmdRegNotMitigationLvl, 1548
- tmd\_mitigationDevList, 620
  - maxMitigationLevel, 621
  - mitigationDevId, 621
  - mitigationDevIdLen, 621
- tmrSIPRegn
  - ims\_SIPRegnTmrInfo, 94
- tmrT1
  - ims\_TmrT1Info, 96
- tmrT2
  - ims\_TmrT2Info, 96
- tmrTf
  - ims\_TmrTfInfo, 97

- toServiceId
  - sms\_BroadcastConfig, [595](#)
- toggleMode
  - voice\_lineCtrlInfo, [1145](#)
- token
  - uim\_indToken, [638](#)
- TokenBucket
  - unpack\_qos\_swiQosFlow\_t, [926](#)
- tokenRate
  - unpack\_qos\_tokenBucket\_t, [927](#)
- tosMask
  - LibPackTFTIDParams, [144](#)
- total\_rx\_bytes
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- total\_rx\_pkt
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- total\_tx\_bytes
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- total\_tx\_bytes\_drp
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- total\_tx\_pkt
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- total\_tx\_pkt\_drp
  - unpack\_qos\_SLQSQosSwiReadDataStats\_t, [915](#)
- TrackAreaCode
  - unpack\_nas\_SLQSGetServingSystem\_t, [826](#)
  - unpack\_nas\_SLQSGetServingSystemV2\_t, [831](#)
- TrackingStatus
  - unpack\_pds\_SetPdsState\_Ind\_t, [900](#)
- TrafficClass
  - unpack\_qos\_swiQosFlow\_t, [926](#)
- trafficClass
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMinQoS, [1179](#)
- trafficPriority
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMinQoS, [1179](#)
- TranDstPort
  - unpack\_qos\_swiQosFilter\_t, [922](#)
- TranSrcPort
  - unpack\_qos\_swiQosFilter\_t, [922](#)
- TransCap
  - sms\_transLayerInfo, [607](#)
- TransType
  - sms\_transLayerInfo, [608](#)
- transactionID
  - sMSTransferRouteMTMessageInfo, [613](#)
- transferDelay
  - LibPackUMTSQoS, [146](#)
  - wds\_UMTSMinQoS, [1179](#)
- TransferRouteMTMessageInfo
  - transferRouteMessageTlv, [621](#)
- transferRouteMessageTlv, [621](#)
  - TlvPresent, [621](#)
  - TransferRouteMTMessageInfo, [621](#)
- transferStats
  - pack\_wds\_SLQSSetWdsEventCallback\_t, [575](#)
- trueSrvStatus
  - nas\_GSMSrvStatusInfo, [210](#)
  - nas\_NR5GSerStatTlv, [268](#)
  - nas\_NR5GSrvStatusTlv, [269](#)
- tx\_bytes
  - unpack\_QosFlowStat\_t, [928](#)
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- tx\_bytes\_drp
  - unpack\_QosFlowStat\_t, [928](#)
- tx\_pkt
  - unpack\_QosFlowStat\_t, [928](#)
- tx\_pkt\_drp
  - unpack\_QosFlowStat\_t, [928](#)
- tx\_pkts
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- TxByteOKCnt
  - wds\_TXBytesOKTlv, [1177](#)
- TxDropConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, [931](#)
- txOKBytesCount
  - unpack\_wds\_SLQSGetDUNCallInfo\_t, [1080](#)
- TxOkByteCountTlv
  - unpack\_RMTransferStatistics\_ind\_t, [931](#)
- TxOkConutTlv
  - unpack\_RMTransferStatistics\_ind\_t, [931](#)
- txPower
  - nas\_txInfo, [335](#)
- TxQFilter
  - unpack\_qos\_QosFlowInfo\_t, [911](#)
- TxQFlowGranted
  - unpack\_qos\_QosFlowInfo\_t, [911](#)
- type
  - pack\_wds\_GetDefaultProfileNum\_t, [556](#)
  - pack\_wds\_SetDefaultProfileNum\_t, [563](#)
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, [704](#)
  - unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, [705](#)
  - unpack\_qmi\_t, [906](#)
- typeSize
  - unpack\_sms\_GetSMSCAddress\_t, [932](#)
- u16PRLVersion
  - unpack\_dms\_GetPRLVersion\_t, [689](#)
- u8PRLPreference
  - unpack\_dms\_GetPRLVersion\_t, [689](#)
- UDPDstPort
  - unpack\_qos\_swiQosFilter\_t, [922](#)
- UDPSrcPort
  - unpack\_qos\_swiQosFilter\_t, [922](#)
- ULONG
  - SwiDataTypes.h, [1517](#)
- ULONGLONG
  - SwiDataTypes.h, [1517](#)
- UMTSGrantedQoS
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, [1084](#)
- UMTSInstInfo
  - nas\_UMTSInfo, [339](#)
- UMTSLTENbrCell



- nas\_WCDMAInfoLTENeighborCell, 347
- UMTSReqQoS
  - LibPackUMTSReqQoSSigInd, 147
- UNIQUE\_ID\_LEN
  - dms.h, 1208
- UNUSEDPARAM
  - common.h, 1195
  - SwiDataTypes.h, 1517
- uPhyCardStatus
  - slot\_t, 593
- uPhySlotStatus
  - slot\_t, 593
- USBComp
  - pack\_dms\_SetUSBComp\_t, 388
  - unpack\_dms\_GetUSBComp\_t, 692
- USHORT
  - SwiDataTypes.h, 1517
- USSDNotificationNetworkInfo
  - unpack\_voice\_USSDNotificationCallback\_ind\_t, 1044
- USSInformation
  - pack\_voice\_SLQSVoiceOrigUSSDNoWait\_t, 548
- UUSData
  - voice\_UUSInfo, 1157
- UUSDatalen
  - voice\_UUSInfo, 1157
- UUSDcs
  - voice\_UUSInfo, 1157
- UUSType
  - voice\_UUSInfo, 1157
- uarfcn
  - nas\_lteWcdmaCellInfo, 254
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
  - nas\_wcdmaUARFCN, 352
- uelIdle
  - nas\_LTEInfoInterfreq, 232
  - nas\_LTEInfoIntrafreq, 234
  - nas\_LTEInfoNeighboringGSM, 235
  - nas\_LTEInfoNeighboringWCDMA, 236
- uim.h, 1549
  - MAX\_ATR\_LENGTH, 1553
  - MAX\_ICCID\_LENGTH, 1553
  - MAX\_NO\_OF\_SLOTS, 1553
  - MAX\_PHY\_SLOTS\_INFO, 1553
  - MAX\_SLOTS\_STATUS, 1553
  - pack\_uim\_ChangePin, 1554
  - pack\_uim\_GetCardStatus, 1554
  - pack\_uim\_ReadTransparent, 1554
  - pack\_uim\_SLQSUIAuthenticate, 1555
  - pack\_uim\_SLQSUIDepersonalization, 1556
  - pack\_uim\_SLQSUIEventRegister, 1556
  - pack\_uim\_SLQSUIGetConfiguration, 1556
  - pack\_uim\_SLQSUIGetFileAttributes, 1557
  - pack\_uim\_SLQSUIGetServiceStatus, 1557
  - pack\_uim\_SLQSUIGetSlotsStatus, 1557
  - pack\_uim\_SLQSUIPowerDown, 1558
  - pack\_uim\_SLQSUIPowerUp, 1558
  - pack\_uim\_SLQSUIReadRecord, 1559
  - pack\_uim\_SLQSUIRefreshComplete, 1559
  - pack\_uim\_SLQSUIRefreshGetLastEvent, 1559
  - pack\_uim\_SLQSUIRefreshOK, 1560
  - pack\_uim\_SLQSUIRefreshRegister, 1560
  - pack\_uim\_SLQSUIReset, 1560
  - pack\_uim\_SLQSUISetServiceStatus, 1561
  - pack\_uim\_SLQSUISwitchSlot, 1561
  - pack\_uim\_SLQSUIWriteRecord, 1561
  - pack\_uim\_SLQSUIWriteTransparent, 1562
  - pack\_uim\_SetPinProtection, 1555
  - pack\_uim\_UnblockPin, 1562
  - pack\_uim\_VerifyPin, 1563
  - unpack\_uim\_ChangePin, 1563
  - unpack\_uim\_GetCardStatus, 1563
  - unpack\_uim\_GetCardStatusV2, 1564
  - unpack\_uim\_ReadTransparent, 1564
  - unpack\_uim\_SLQSUIAuthenticate, 1565
  - unpack\_uim\_SLQSUIDepersonalization, 1565
  - unpack\_uim\_SLQSUIEventRegister, 1566
  - unpack\_uim\_SLQSUIGetConfiguration, 1566
  - unpack\_uim\_SLQSUIGetFileAttributes, 1566
  - unpack\_uim\_SLQSUIGetServiceStatus, 1567
  - unpack\_uim\_SLQSUIGetSlotsStatus, 1567
  - unpack\_uim\_SLQSUIGetSlotsStatusV2, 1568
  - unpack\_uim\_SLQSUIPowerDown, 1568
  - unpack\_uim\_SLQSUIPowerDown\_t, 1553
  - unpack\_uim\_SLQSUIPowerUp, 1568
  - unpack\_uim\_SLQSUIPowerUp\_t, 1553
  - unpack\_uim\_SLQSUIReadRecord, 1569
  - unpack\_uim\_SLQSUIRefreshCallback\_Ind, 1569
  - unpack\_uim\_SLQSUIRefreshComplete, 1569
  - unpack\_uim\_SLQSUIRefreshComplete\_t, 1554
  - unpack\_uim\_SLQSUIRefreshGetLastEvent, 1570
  - unpack\_uim\_SLQSUIRefreshOK, 1570
  - unpack\_uim\_SLQSUIRefreshOK\_t, 1554
  - unpack\_uim\_SLQSUIRefreshRegister, 1570
  - unpack\_uim\_SLQSUIRefreshRegister\_t, 1554
  - unpack\_uim\_SLQSUIReset, 1571
  - unpack\_uim\_SLQSUIReset\_t, 1554
  - unpack\_uim\_SLQSUISetServiceStatus, 1571
  - unpack\_uim\_SLQSUISetServiceStatus\_t, 1554
  - unpack\_uim\_SLQSUISetStatusChangeCall-Back\_ind, 1571
  - unpack\_uim\_SLQSUISwitchSlot, 1572
  - unpack\_uim\_SLQSUISwitchSlot\_t, 1554
  - unpack\_uim\_SLQSUIWriteRecord, 1572
  - unpack\_uim\_SLQSUIWriteTransparent, 1572
  - unpack\_uim\_SetPinProtection, 1564
  - unpack\_uim\_SetUimSlotStatusChangeCallback\_ind, 1565
  - unpack\_uim\_UnblockPin, 1573
  - unpack\_uim\_UnblockPinV2, 1573
  - unpack\_uim\_VerifyPin, 1573
- uim\_GetSlotsInfoTlv, 635
- NumberOfPhySlotInfo, 636
- TlvPresent, 636

- uimSlotInfo, 636
- uim\_GetSlotsStatusTlv, 636
  - NumberOfPhySlot, 636
  - TlvPresent, 636
- uimSlotStatus, 636
- uim\_UIMGetFDNStatus, 649
  - FDNStatus, 650
  - TlvPresent, 650
- uim\_UIMGetHiddenKeyStatus, 650
  - hiddenKey, 650
  - TlvPresent, 650
- uim\_UIMGetIndex, 650
  - index, 651
  - TlvPresent, 651
- uim\_UIMSessionInformation, 651
  - aid, 651
  - aidLength, 652
  - sessionType, 652
- uim\_additionalReadResult, 621
  - additionalRecord, 622
  - additionalRecordLen, 622
  - TlvPresent, 622
- uim\_appStatus, 622
  - aidLength, 624
  - aidVal, 624
  - appState, 624
  - appType, 625
  - persoFeature, 625
  - persoRetries, 625
  - persoState, 625
  - persoUnblockRetries, 625
  - pin1Retries, 625
  - pin1State, 625
  - pin2Retries, 625
  - pin2State, 625
  - puk1Retries, 625
  - puk2Retries, 625
  - univPin, 625
- uim\_authenticateResult, 625
  - content, 625
  - contentLen, 625
- uim\_authenticationData, 626
  - context, 626
  - data, 626
  - dataLen, 626
- uim\_cardResult, 627
  - sw1, 627
  - sw2, 627
- uim\_cardResultInfo, 627
  - sw1, 627
  - sw2, 628
  - TlvPresent, 628
- uim\_cardStatus, 628
  - index1xPri, 629
  - index1xSec, 629
  - indexGwPri, 629
  - indexGwSec, 629
  - numSlot, 629
  - SlotInfo, 629
- uim\_changeUIMPIN, 629
  - oldPINLen, 629
  - oldPINVal, 630
  - pinID, 630
  - pinLen, 630
  - pinVal, 630
- uim\_depersonalizationInformation, 630
  - ckLen, 631
  - ckVal, 631
  - feature, 631
  - operation, 631
- uim\_encryptedPIN1, 631
  - pin1Len, 631
  - pin1Val, 631
- uim\_fileAttributes, 631
  - fileID, 634
  - fileSize, 634
  - fileType, 634
  - rawLen, 634
  - rawValue, 634
  - recordCount, 634
  - recordSize, 634
  - secActivate, 634
  - secActivateMask, 634
  - secDeactivate, 634
  - secDeactivateMask, 634
  - secIncrease, 634
  - secIncreaseMask, 634
  - secRead, 634
  - secReadMask, 634
  - secWrite, 635
  - secWriteMask, 635
- uim\_fileInfo, 635
  - fileID, 635
  - path, 635
  - pathLen, 635
- uim\_hotSwapStatus, 637
  - hotSwap, 637
  - hotSwapLength, 637
- uim\_indToken, 637
  - TlvPresent, 637
  - token, 638
- uim\_personalizationStatus, 638
  - feature, 638
  - numFeatures, 638
  - unblockLeft, 638
  - verifyLeft, 638
- uim\_physlotInfo, 639
  - atrValue, 639
  - atrValueLen, 639
  - cardProtocol, 639
  - iseUICC, 639
  - numApp, 639
- uim\_physlotsInfo, 639
  - uimSlotInfo, 640
- uim\_readRecordInfo, 640
  - length, 640

- record, 640
- uim\_readResult, 640
  - content, 641
  - contentLen, 641
- uim\_readResultInfo, 641
  - content, 641
  - contentLen, 641
  - TlvPresent, 641
- uim\_readTransparentInfo, 642
  - length, 642
  - offset, 642
- uim\_refreshevent, 642
  - aid, 643
  - aidLength, 643
  - arrfileInfo, 643
  - mode, 643
  - numOfFiles, 643
  - sessionType, 644
  - stage, 644
- uim\_registerRefresh, 644
  - arrfileInfo, 644
  - numFiles, 644
  - registerFlag, 644
  - voteForInit, 644
- uim\_remainingRetries, 645
  - unblockLeft, 645
  - verifyLeft, 645
- uim\_select
  - pack\_dms\_SwiUimSelect\_t, 393
- uim\_sessionInformation, 645
  - aid, 646
  - aidLength, 646
  - sessionType, 646
- uim\_setPINProtection, 646
  - pinID, 647
  - pinLength, 647
  - pinOperation, 647
  - pinValue, 647
- uim\_simBusyStatus, 647
  - simBusy, 647
  - simBusyLength, 647
- uim\_slotInfo, 648
  - AppStatus, 649
  - cardState, 649
  - errorState, 649
  - numApp, 649
  - upinRetries, 649
  - upinState, 649
  - upukRetries, 649
- uim\_unblockUIMPIN, 652
  - newPINLen, 652
  - newPINVal, 652
  - pinID, 652
  - pukLen, 652
  - pukVal, 652
- uim\_validCardStatus, 653
  - validCard, 653
  - validCardLength, 653
- uim\_verifyUIMPIN, 653
  - pinID, 654
  - pinLen, 654
  - pinVal, 654
- uim\_writeRecordInfo, 654
  - data, 654
  - dataLen, 654
  - record, 654
- uim\_writeTransparentInfo, 654
  - data, 655
  - dataLen, 655
  - offset, 655
- uimAutoSwitchActSlot
  - dms\_UimAutoSwitchActSlotTlv, 72
- uimSelect
  - unpack\_dms\_SLQSDmsSwiGetUimSelection\_t, 706
- uimSlotInfo
  - uim\_GetSlotsInfoTlv, 636
  - uim\_physlotsInfo, 640
- uimSlotStatus
  - slots\_t, 595
  - uim\_GetSlotsStatusTlv, 636
- UimStatusTlv
  - unpack\_dms\_SwiEventReportCallBack\_ind\_t, 729
- ulData
  - wds\_DataULongTlv, 1162
- ulMask
  - rmTrasferStaticsReq, 590
- ulMax
  - PackSwiAvmsSetSettingsPeriodInfo, 584
- ulMin
  - PackSwiAvmsSetSettingsPeriodInfo, 584
- ulPhysicalSlot
  - pack\_uim\_SLQSUIMSwitchSlot\_t, 532
- ulPkgDownloadComplete
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- ulPkgDownloadSize
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- ulldata
  - wds\_DataULongLongTlv, 1162
- umtsEcio
  - nas\_umtsInstArr, 340
  - nas\_UMTSinstInfo, 341
- umtsInst
  - nas\_UMTSExtInfo, 337
  - nas\_UMTSInfo, 339
- umtsInstElement
  - nas\_UMTSExtInfo, 337
- umtsLTENbrCellLen
  - nas\_WCDMAInfoLTENeighborCell, 347
- umtsPsc
  - nas\_umtsInstArr, 340
  - nas\_UMTSinstInfo, 341
- umtsRank
  - nas\_umtsInstArr, 340



- umtsRscp
  - nas\_umtsInstArr, 340
  - nas\_UMTSinstInfo, 341
- umtsSet
  - nas\_umtsInstArr, 340
- umtsSqual
  - nas\_umtsInstArr, 340
- umtsSrxLvl
  - nas\_umtsInstArr, 340
- umtsUarfcn
  - nas\_umtsInstArr, 340
  - nas\_UMTSinstInfo, 341
- UnPackGetProfileSettingOut, 1099
  - curProfile, 1099
  - pExtErrCode, 1099
- UnPackGetProfileSettingOutV2, 1099
  - curProfile, 1100
  - pExtErrCode, 1100
- unblockLeft
  - uim\_personalizationStatus, 638
  - uim\_remainingRetries, 645
- unblockRetriesLeft
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 731
  - unpack\_dms\_UIMSetPINProtection\_t, 735
  - unpack\_dms\_UIMUnblockControlKey\_t, 735
- uniqueID
  - image\_info\_t, 84
- univPin
  - appStats, 42
  - uim\_appStatus, 625
- universalTime
  - unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t, 870
- unpack\_QosFlowStat\_t, 927
  - bearerId, 928
  - tx\_bytes, 928
  - tx\_bytes\_drp, 928
  - tx\_pkt, 928
  - tx\_pkt\_drp, 928
- unpack\_RMTransferStatistics\_ind\_t, 930
  - ParamPresenceMask, 930
  - RxDropConutTlv, 930
  - RxOkByteCountTlv, 930
  - RxOkConutTlv, 931
  - TxDropConutTlv, 931
  - TxOkByteCountTlv, 931
  - TxOkConutTlv, 931
- unpack\_audio\_SLQSGetAudioPathConfig
  - audio.h, 1187
- unpack\_audio\_SLQSGetAudioPathConfig\_t, 655
  - pCodecSTGain, 657
  - pECMode, 657
  - pMICGainSelect, 657
  - pNSEnable, 657
  - pRXAGCList, 657
  - pRXAVCList, 657
  - pTXAGCList, 657
  - pTXAVCSwitch, 657
  - pTXGain, 657
  - ParamPresenceMask, 657
- unpack\_audio\_SLQSGetAudioProfile
  - audio.h, 1187
- unpack\_audio\_SLQSGetAudioProfile\_t, 657
  - EarMute, 658
  - MicMute, 658
  - ParamPresenceMask, 658
  - Profile, 658
  - Volume, 658
- unpack\_audio\_SLQSGetAudioVolTLBConfig
  - audio.h, 1187
- unpack\_audio\_SLQSGetAudioVolTLBConfig\_t, 658
  - ParamPresenceMask, 659
  - ResCode, 659
- unpack\_audio\_SLQSSetAudioPathConfig
  - audio.h, 1188
- unpack\_audio\_SLQSSetAudioProfile
  - audio.h, 1188
- unpack\_audio\_SLQSSetAudioVolTLBConfig
  - audio.h, 1188
- unpack\_audio\_SLQSSetAudioVolTLBConfig\_t, 659
  - ParamPresenceMask, 659
  - ResCode, 659
- unpack\_cat\_CATSendEnvelopeCommand
  - cat.h, 1191
- unpack\_cat\_CATSendEnvelopeCommand\_t
  - cat.h, 1190
- unpack\_cat\_CATSendTerminalResponse
  - cat.h, 1191
- unpack\_cat\_CATSendTerminalResponse\_t
  - cat.h, 1190
- unpack\_cat\_SetCATEventCallback
  - cat.h, 1192
- unpack\_cat\_SetCATEventCallback\_t, 660
  - errorMask, 661
  - ParamPresenceMask, 661
  - Tlvresult, 661
- unpack\_cat\_SetCatEventCallback\_ind
  - cat.h, 1192
- unpack\_cat\_SetCatEventCallback\_ind\_t, 659
  - CCETlv, 660
  - event\_Index, 660
  - ParamPresenceMask, 660
- unpack\_dms\_ActivateAutomatic
  - dms.h, 1233
- unpack\_dms\_GetActivationState
  - dms.h, 1234
- unpack\_dms\_GetActivationState\_t, 661
  - ParamPresenceMask, 662
  - state, 662
- unpack\_dms\_GetBandCapability
  - dms.h, 1234
- unpack\_dms\_GetBandCapability\_t, 662
  - BandCapability, 664
  - ParamPresenceMask, 664
  - Tlvresult, 664
- unpack\_dms\_GetCrashAction

- dms.h, [1234](#)
- unpack\_dms\_GetCrashAction\_t, [664](#)
  - DevCrashState, [664](#)
  - ParamPresenceMask, [664](#)
  - Tlvresult, [665](#)
- unpack\_dms\_GetCustFeature
  - dms.h, [1235](#)
- unpack\_dms\_GetCustFeature\_t, [665](#)
  - DHCPRelayEnabled, [667](#)
  - DisableIMSI, [667](#)
  - GPSPMP, [667](#)
  - GPSSel, [667](#)
  - GpsEnable, [667](#)
  - IPFamSupport, [667](#)
  - IsVoiceEnabled, [667](#)
  - ParamPresenceMask, [667](#)
  - RMAutoConnect, [667](#)
  - SMSSupport, [667](#)
  - Tlvresult, [667](#)
- unpack\_dms\_GetCustFeaturesV2
  - dms.h, [1235](#)
- unpack\_dms\_GetCustFeaturesV2\_t, [667](#)
  - GetCustomFeatureV2, [668](#)
  - ParamPresenceMask, [668](#)
  - Tlvresult, [668](#)
- unpack\_dms\_GetDeviceCap
  - dms.h, [1235](#)
- unpack\_dms\_GetDeviceCap\_t, [668](#)
  - DataServiceCapability, [669](#)
  - MaxRXChannelRate, [669](#)
  - MaxTXChannelRate, [669](#)
  - ParamPresenceMask, [669](#)
  - Radiofaces, [669](#)
  - RadiofacesSize, [669](#)
  - SimCapability, [669](#)
  - Tlvresult, [669](#)
- unpack\_dms\_GetDeviceCapabilities
  - dms.h, [1236](#)
- unpack\_dms\_GetDeviceCapabilities\_t, [669](#)
  - dataServiceCaCapability, [670](#)
  - maxRxChannelRate, [671](#)
  - maxTxChannelRate, [671](#)
  - ParamPresenceMask, [671](#)
  - Radiofaces, [671](#)
  - radiofacesSize, [671](#)
  - simCapability, [671](#)
- unpack\_dms\_GetDeviceCapabilitiesV2
  - dms.h, [1236](#)
- unpack\_dms\_GetDeviceCapabilitiesV2\_t, [671](#)
  - DevCaps, [673](#)
  - pDevCurSubsCaps, [673](#)
  - pDevExplicitCfgIndex, [673](#)
  - pDevMaxActDataSubsCaps, [673](#)
  - pDevMaxCfgListCaps, [673](#)
  - pDevMaxSubsCaps, [673](#)
  - pDevMultiSimCaps, [673](#)
  - pDevMultiSimVoiceDataCaps, [673](#)
  - pDevSrvCaps, [673](#)
  - pDevSubsFeatureModeCaps, [673](#)
  - pDevSubsVoiceDataCaps, [673](#)
  - pDevVoiceCaps, [673](#)
  - pDevVoiceDataCaps, [673](#)
  - ParamPresenceMask, [673](#)
  - Tlvresult, [673](#)
- unpack\_dms\_GetDeviceHardwareRev
  - dms.h, [1236](#)
- unpack\_dms\_GetDeviceHardwareRev\_t, [673](#)
  - ParamPresenceMask, [674](#)
  - String, [674](#)
  - stringSize, [674](#)
  - Tlvresult, [674](#)
- unpack\_dms\_GetDeviceMfr
  - dms.h, [1237](#)
- unpack\_dms\_GetDeviceMfr\_t, [674](#)
  - ParamPresenceMask, [675](#)
  - String, [675](#)
  - stringSize, [675](#)
  - Tlvresult, [675](#)
- unpack\_dms\_GetDeviceSerialNumbers
  - dms.h, [1237](#)
- unpack\_dms\_GetDeviceSerialNumbers\_t, [675](#)
  - ESNString, [677](#)
  - esnSize, [676](#)
  - IMEIString, [677](#)
  - imeiSize, [677](#)
  - imeiSvnSize, [677](#)
  - ImeiSvnString, [677](#)
  - MEIDString, [677](#)
  - meidSize, [677](#)
  - ParamPresenceMask, [677](#)
  - Tlvresult, [677](#)
- unpack\_dms\_GetFSN
  - dms.h, [1238](#)
- unpack\_dms\_GetFSN\_t, [681](#)
  - ParamPresenceMask, [681](#)
  - String, [681](#)
  - Tlvresult, [681](#)
- unpack\_dms\_GetFirmwareInfo
  - dms.h, [1237](#)
- unpack\_dms\_GetFirmwareInfo\_t, [677](#)
  - appversion\_str, [678](#)
  - bootversion\_str, [678](#)
  - carrier\_str, [678](#)
  - cur\_carr\_name, [678](#)
  - cur\_carr\_rev, [678](#)
  - modelid\_str, [678](#)
  - packageid\_str, [678](#)
  - ParamPresenceMask, [678](#)
  - priversion\_str, [678](#)
  - sku\_str, [678](#)
  - Tlvresult, [678](#)
- unpack\_dms\_GetFirmwareRevision
  - dms.h, [1238](#)
- unpack\_dms\_GetFirmwareRevision\_t, [679](#)
  - AMSSString, [679](#)
  - amssSize, [679](#)

- PRIStrng, [679](#)
- ParamPresenceMask, [679](#)
- Tlvresult, [679](#)
- unpack\_dms\_GetFirmwareRevisions
  - dms.h, [1238](#)
- unpack\_dms\_GetFirmwareRevisions\_t, [680](#)
  - AMSSString, [681](#)
  - amssSize, [681](#)
  - bootSize, [681](#)
  - BootString, [681](#)
  - PRIStrng, [681](#)
  - ParamPresenceMask, [681](#)
  - priSize, [681](#)
  - Tlvresult, [681](#)
- unpack\_dms\_GetHardwareRevision
  - dms.h, [1239](#)
- unpack\_dms\_GetHardwareRevision\_t, [682](#)
  - hwVer, [682](#)
  - ParamPresenceMask, [682](#)
- unpack\_dms\_GetIMSI
  - dms.h, [1239](#)
- unpack\_dms\_GetIMSI\_t, [682](#)
  - imsi, [682](#)
  - ParamPresenceMask, [683](#)
  - Tlvresult, [683](#)
- unpack\_dms\_GetManufacturer
  - dms.h, [1239](#)
- unpack\_dms\_GetManufacturer\_t, [683](#)
  - manufacturer, [683](#)
  - ParamPresenceMask, [683](#)
  - Tlvresult, [683](#)
- unpack\_dms\_GetModelID
  - dms.h, [1240](#)
- unpack\_dms\_GetModelID\_t, [683](#)
  - modelid, [684](#)
  - ParamPresenceMask, [684](#)
  - Tlvresult, [684](#)
- unpack\_dms\_GetNetworkTime
  - dms.h, [1240](#)
- unpack\_dms\_GetNetworkTime\_t, [684](#)
  - ParamPresenceMask, [685](#)
  - source, [685](#)
  - timestamp, [685](#)
  - Tlvresult, [685](#)
- unpack\_dms\_GetNetworkTimeV2
  - dms.h, [1240](#)
- unpack\_dms\_GetNetworkTimeV2\_t, [685](#)
  - pSysTime, [686](#)
  - pUsrTime, [686](#)
  - ParamPresenceMask, [686](#)
  - source, [686](#)
  - timestamp, [686](#)
  - Tlvresult, [686](#)
- unpack\_dms\_GetOfflineReason
  - dms.h, [1241](#)
- unpack\_dms\_GetOfflineReason\_t, [686](#)
  - pReasonMask, [687](#)
  - ParamPresenceMask, [687](#)
- pbPlatform, [687](#)
- Tlvresult, [687](#)
- unpack\_dms\_GetPRLVersion
  - dms.h, [1241](#)
- unpack\_dms\_GetPRLVersion\_t, [688](#)
  - ParamPresenceMask, [689](#)
  - Tlvresult, [689](#)
  - u16PRLVersion, [689](#)
  - u8PRLPreference, [689](#)
- unpack\_dms\_GetPower
  - dms.h, [1241](#)
- unpack\_dms\_GetPower\_t, [687](#)
  - HardwareControlledMode, [688](#)
  - OfflineReason, [688](#)
  - OperationMode, [688](#)
  - ParamPresenceMask, [688](#)
  - Tlvresult, [688](#)
- unpack\_dms\_GetSerialNumbers
  - dms.h, [1242](#)
- unpack\_dms\_GetSerialNumbers\_t, [689](#)
  - esn, [690](#)
  - imei\_no, [690](#)
  - imeisv\_svn, [690](#)
  - meid, [690](#)
  - ParamPresenceMask, [690](#)
- unpack\_dms\_GetUSBComp
  - dms.h, [1242](#)
- unpack\_dms\_GetUSBComp\_t, [690](#)
  - NumSupUSBComps, [692](#)
  - ParamPresenceMask, [692](#)
  - SupUSBComps, [692](#)
  - Tlvresult, [692](#)
  - USBComp, [692](#)
- unpack\_dms\_GetVoiceNumber
  - dms.h, [1242](#)
- unpack\_dms\_GetVoiceNumber\_t, [692](#)
  - MIN, [693](#)
  - minSize, [693](#)
  - ParamPresenceMask, [693](#)
  - Tlvresult, [693](#)
  - VoiceNumber, [693](#)
  - voiceNumberSize, [693](#)
- unpack\_dms\_PSMCfgChange\_ind
  - dms.h, [1243](#)
- unpack\_dms\_PSMCfgChange\_ind\_t, [694](#)
  - ActiveTimer, [694](#)
  - EnableState, [694](#)
  - ParamPresenceMask, [694](#)
  - PeriodicUpdateTimer, [694](#)
  - Tlvresult, [694](#)
- unpack\_dms\_ResetToFactoryDefaults
  - dms.h, [1243](#)
- unpack\_dms\_ResetToFactoryDefaults\_t, [694](#)
  - ParamPresenceMask, [695](#)
  - Tlvresult, [695](#)
- unpack\_dms\_SLQSDmsSwiGetPCInfo
  - dms.h, [1247](#)
- unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, [700](#)

- has\_LpmFlag, 702
- has\_PersistentLpm, 702
- has\_Wdisable, 702
- LpmFlag, 703
- opMode, 703
- ParamPresenceMask, 703
- PersistentLpm, 703
- PowerOffMode, 703
- Wdisable, 703
- unpack\_dms\_SLQSDmsSwiGetResetInfo
  - dms.h, 1247
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind
  - dms.h, 1248
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_Ind\_t, 703
  - ParamPresenceMask, 704
  - source, 704
  - Tlvresult, 704
  - type, 704
- unpack\_dms\_SLQSDmsSwiGetResetInfo\_t, 704
  - ParamPresenceMask, 705
  - source, 705
  - Tlvresult, 705
  - type, 705
- unpack\_dms\_SLQSDmsSwiGetUimSelection
  - dms.h, 1248
- unpack\_dms\_SLQSDmsSwiGetUimSelection\_t, 705
  - pUimAutoSwitchActSlot, 706
  - ParamPresenceMask, 706
  - uimSelect, 706
- unpack\_dms\_SLQSDmsSwiIndicationRegister
  - dms.h, 1248
- unpack\_dms\_SLQSDmsSwiIndicationRegister\_t, 706
  - ParamPresenceMask, 707
  - Tlvresult, 707
- unpack\_dms\_SLQSGetBandCapability
  - dms.h, 1249
- unpack\_dms\_SLQSGetBandCapability\_t, 707
  - bandCapability, 711
  - LteBandCapability, 711
  - ParamPresenceMask, 711
  - TdsBandCapability, 711
- unpack\_dms\_SLQSGetBandCapabilityExt
  - dms.h, 1249
- unpack\_dms\_SLQSGetBandCapabilityExt\_t, 711
  - bandCapability, 715
  - LteBandCapability, 715
  - LteBandsSupport, 715
  - ParamPresenceMask, 715
  - TdsBandCapability, 715
- unpack\_dms\_SLQSGetERIFile
  - dms.h, 1250
- unpack\_dms\_SLQSGetERIFile\_t, 715
  - eriFile, 716
  - ParamPresenceMask, 716
  - Tlvresult, 716
- unpack\_dms\_SLQSGetPowerSaveModeConfig
  - dms.h, 1250
- unpack\_dms\_SLQSGetPowerSaveModeConfig\_t, 716
  - pActiveTimer, 717
  - pDurationThreshold, 717
  - pEarlyWakeupTime, 717
  - pPeriodicUpdateTimer, 717
  - pPsmEnableState, 717
  - pRandomizationWindow, 717
  - ParamPresenceMask, 717
- unpack\_dms\_SLQSSetPowerSaveModeConfig
  - dms.h, 1250
- unpack\_dms\_SLQSSetPowerSaveModeConfig\_t, 717
  - ParamPresenceMask, 718
  - Tlvresult, 718
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics
  - dms.h, 1251
- unpack\_dms\_SLQSSwiClearDyingGaspStatistics\_t, 718
  - ParamPresenceMask, 718
  - Tlvresult, 718
- unpack\_dms\_SLQSSwiGetCrashInfo
  - dms.h, 1251
- unpack\_dms\_SLQSSwiGetCrashInfo\_t, 718
  - crashInfoParam, 719
  - ParamPresenceMask, 719
  - Tlvresult, 719
- unpack\_dms\_SLQSSwiGetDyingGaspCfg
  - dms.h, 1251
- unpack\_dms\_SLQSSwiGetDyingGaspCfg\_t, 719
  - pGetDyingGaspCfg, 720
  - ParamPresenceMask, 720
  - Tlvresult, 720
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics
  - dms.h, 1252
- unpack\_dms\_SLQSSwiGetDyingGaspStatistics\_t, 720
  - pGetDyingGaspStatistics, 720
  - ParamPresenceMask, 720
  - Tlvresult, 720
- unpack\_dms\_SLQSSwiGetFirmwareCurr
  - dms.h, 1252
- unpack\_dms\_SLQSSwiGetFirmwareCurr\_t, 720
  - carrier, 721
  - fwvers, 721
  - numEntries, 721
  - pCurrImgInfo, 721
  - ParamPresenceMask, 721
  - pkgver, 721
  - priver, 721
- unpack\_dms\_SLQSSwiGetFwUpdateStatus
  - dms.h, 1252
- unpack\_dms\_SLQSSwiGetFwUpdateStatus\_t, 721
  - imgType, 723
  - logString, 723
  - ParamPresenceMask, 723
  - refData, 723
  - refString, 723
  - ResCode, 723
  - Tlvresult, 723
- unpack\_dms\_SLQSSwiGetHostDevInfo
  - dms.h, 1253

- unpack\_dms\_SLQSSwiGetHostDevInfo\_t, 723
  - hostID, 724
  - manString, 724
  - modelString, 724
  - ParamPresenceMask, 724
  - plasmaIDString, 724
  - swVerString, 724
  - Tlvresult, 724
- unpack\_dms\_SLQSSwiGetOSInfo
  - dms.h, 1253
- unpack\_dms\_SLQSSwiGetOSInfo\_t, 724
  - nameString, 725
  - ParamPresenceMask, 725
  - Tlvresult, 725
  - versionString, 725
- unpack\_dms\_SLQSSwiGetSerialNoExt
  - dms.h, 1253
- unpack\_dms\_SLQSSwiGetSerialNoExt\_t, 725
  - meidString, 726
  - ParamPresenceMask, 726
  - Tlvresult, 726
- unpack\_dms\_SLQSSwiSetDyingGaspCfg
  - dms.h, 1254
- unpack\_dms\_SLQSSwiSetDyingGaspCfg\_t, 726
  - ParamPresenceMask, 726
  - Tlvresult, 726
- unpack\_dms\_SLQSSwiSetHostDevInfo
  - dms.h, 1254
- unpack\_dms\_SLQSSwiSetHostDevInfo\_t, 726
  - ParamPresenceMask, 727
  - Tlvresult, 727
- unpack\_dms\_SLQSSwiSetOSInfo
  - dms.h, 1254
- unpack\_dms\_SLQSSwiSetOSInfo\_t, 727
  - ParamPresenceMask, 727
  - Tlvresult, 727
- unpack\_dms\_SLQSUIMGetState
  - dms.h, 1255
- unpack\_dms\_SLQSUIMGetState\_t, 727
  - ParamPresenceMask, 728
  - state, 728
  - Tlvresult, 728
- unpack\_dms\_SetActivationStatusCallback
  - dms.h, 1244
- unpack\_dms\_SetActivationStatusCallback\_t, 695
  - ParamPresenceMask, 695
  - Tlvresult, 695
- unpack\_dms\_SetCrashAction
  - dms.h, 1244
- unpack\_dms\_SetCrashAction\_t, 695
  - notused, 696
  - ParamPresenceMask, 696
- unpack\_dms\_SetCustFeature
  - dms.h, 1244
- unpack\_dms\_SetCustFeature\_t, 696
  - ParamPresenceMask, 696
  - Tlvresult, 696
- unpack\_dms\_SetCustFeaturesV2
  - dms.h, 1245
- unpack\_dms\_SetCustFeaturesV2\_t, 696
  - ParamPresenceMask, 697
  - Tlvresult, 697
- unpack\_dms\_SetEventReport
  - dms.h, 1245
- unpack\_dms\_SetEventReport\_ind
  - dms.h, 1245
- unpack\_dms\_SetEventReport\_ind\_t, 697
  - ActivationStatusTlv, 697
  - OperatingModeTlv, 697
  - ParamPresenceMask, 698
  - Tlvresult, 698
- unpack\_dms\_SetEventReport\_t, 698
  - ParamPresenceMask, 698
  - Tlvresult, 698
- unpack\_dms\_SetFirmwarePreference
  - dms.h, 1246
- unpack\_dms\_SetFirmwarePreference\_t, 698
  - ParamPresenceMask, 699
  - Tlvresult, 699
- unpack\_dms\_SetIndicationRegister
  - dms.h, 1246
- unpack\_dms\_SetIndicationRegister\_t, 699
  - ParamPresenceMask, 699
  - Tlvresult, 699
- unpack\_dms\_SetPower
  - dms.h, 1246
- unpack\_dms\_SetPower\_t, 699
  - ParamPresenceMask, 700
  - Tlvresult, 700
- unpack\_dms\_SetUSBComp
  - dms.h, 1247
- unpack\_dms\_SetUSBComp\_t, 700
  - ParamPresenceMask, 700
  - Tlvresult, 700
- unpack\_dms\_SwiEventReportCallBack\_ind
  - dms.h, 1255
- unpack\_dms\_SwiEventReportCallBack\_ind\_t, 728
  - ParamPresenceMask, 729
  - TempTlv, 729
  - UimStatusTlv, 729
  - VoltTlv, 729
- unpack\_dms\_SwiSetEventReport
  - dms.h, 1255
- unpack\_dms\_SwiSetEventReport\_t, 729
  - ParamPresenceMask, 729
  - Tlvresult, 729
- unpack\_dms\_SwiUimSelect
  - dms.h, 1256
- unpack\_dms\_SwiUimSelect\_t, 729
  - ParamPresenceMask, 730
  - Tlvresult, 730
- unpack\_dms\_UIMChangePIN
  - dms.h, 1256
- unpack\_dms\_UIMGetControlKeyStatus
  - dms.h, 1256
- unpack\_dms\_UIMGetControlKeyStatus\_t, 730

- facilityState, [731](#)
- ParamPresenceMask, [731](#)
- Tlvresult, [731](#)
- unblockRetriesLeft, [731](#)
- verifyRetriesLeft, [731](#)
- unpack\_dms\_UIMGetICCID
  - dms.h, [1257](#)
- unpack\_dms\_UIMGetICCID\_t, [731](#)
  - ParamPresenceMask, [731](#)
  - String, [731](#)
  - stringSize, [731](#)
  - Tlvresult, [731](#)
- unpack\_dms\_UIMGetPINStatus
  - dms.h, [1257](#)
- unpack\_dms\_UIMGetPINStatus\_t, [732](#)
  - p1Status, [733](#)
  - p1UnblockRetriesLeft, [733](#)
  - p1VerifyRetriesLeft, [733](#)
  - p2Status, [733](#)
  - p2UnblockRetriesLeft, [733](#)
  - p2VerifyRetriesLeft, [733](#)
  - ParamPresenceMask, [733](#)
  - Tlvresult, [733](#)
- unpack\_dms\_UIMSetControlKeyProtection
  - dms.h, [1257](#)
- unpack\_dms\_UIMSetControlKeyProtection\_t, [733](#)
  - ParamPresenceMask, [734](#)
  - Tlvresult, [734](#)
  - verifyRetriesLeft, [734](#)
- unpack\_dms\_UIMSetPINProtection
  - dms.h, [1258](#)
- unpack\_dms\_UIMSetPINProtection\_t, [734](#)
  - ParamPresenceMask, [735](#)
  - Tlvresult, [735](#)
  - unblockRetriesLeft, [735](#)
  - verifyRetriesLeft, [735](#)
- unpack\_dms\_UIMUnblockControlKey
  - dms.h, [1258](#)
- unpack\_dms\_UIMUnblockControlKey\_t, [735](#)
  - ParamPresenceMask, [735](#)
  - Tlvresult, [735](#)
  - unblockRetriesLeft, [735](#)
- unpack\_dms\_UIMUnblockPIN
  - dms.h, [1258](#)
- unpack\_dms\_UIMVerifyPIN
  - dms.h, [1259](#)
- unpack\_dms\_ValidateSPC
  - dms.h, [1259](#)
- unpack\_fms\_GetImagesPreference
  - fms.h, [1262](#)
- unpack\_fms\_GetImagesPreference\_t, [735](#)
  - ImageListSize, [736](#)
  - pImageList, [736](#)
  - ParamPresenceMask, [736](#)
  - Tlvresult, [736](#)
- unpack\_fms\_GetStoredImages
  - fms.h, [1262](#)
- unpack\_fms\_GetStoredImages\_t, [736](#)
- imageList, [737](#)
- imagelistSize, [737](#)
- ParamPresenceMask, [737](#)
- Tlvresult, [737](#)
- unpack\_fms\_SetImagesPreference
  - fms.h, [1263](#)
- unpack\_fms\_SetImagesPreference\_t, [737](#)
  - ImageTypes, [737](#)
  - ImageTypesSize, [737](#)
  - ParamPresenceMask, [737](#)
  - Tlvresult, [737](#)
- unpack\_ims\_SLQSGetIMSSMSConfig
  - ims.h, [1270](#)
- unpack\_ims\_SLQSGetIMSSMSConfig\_t, [737](#)
  - pSettingResp, [738](#)
  - ParamPresenceMask, [738](#)
- unpack\_ims\_SLQSGetIMSUserConfig
  - ims.h, [1270](#)
- unpack\_ims\_SLQSGetIMSUserConfig\_t, [738](#)
  - pIMSDomain, [739](#)
  - pSettingResp, [739](#)
  - ParamPresenceMask, [739](#)
- unpack\_ims\_SLQSGetIMSVoIPConfig
  - ims.h, [1271](#)
- unpack\_ims\_SLQSGetIMSVoIPConfig\_t, [739](#)
  - pAmrMode, [741](#)
  - pRingingTimer, [742](#)
  - pSettingResp, [742](#)
  - ParamPresenceMask, [742](#)
- unpack\_ims\_SLQSGetRegMgrConfig
  - ims.h, [1271](#)
- unpack\_ims\_SLQSGetRegMgrConfig\_t, [742](#)
  - pIMSTestMode, [743](#)
  - pPCSCFPort, [743](#)
  - pSettingResp, [743](#)
  - ParamPresenceMask, [743](#)
- unpack\_ims\_SLQSGetSIPConfig
  - ims.h, [1271](#)
- unpack\_ims\_SLQSGetSIPConfig\_t, [743](#)
  - pSIPLocalPort, [744](#)
  - pSettingResp, [744](#)
  - pSigCompEnabled, [744](#)
  - pSubscribeTimer, [744](#)
  - pTimerSIPReg, [744](#)
  - pTimerT1, [744](#)
  - pTimerT2, [744](#)
  - pTimerTf, [744](#)
  - ParamPresenceMask, [744](#)
- unpack\_ims\_SLQSImsConfigIndicationRegister
  - ims.h, [1272](#)
- unpack\_ims\_SLQSImsConfigIndicationRegister\_t
  - ims.h, [1265](#)
- unpack\_ims\_SLQSRegMgrCfgCallBack\_ind
  - ims.h, [1272](#)
- unpack\_ims\_SLQSRegMgrCfgCallBack\_ind\_t, [744](#)
  - PCTlv, [745](#)
  - PNTlv, [745](#)
  - ParamPresenceMask, [745](#)



- TMTlv, [745](#)
- unpack\_ims\_SLQSSIPCfgCallback\_ind
  - ims.h, [1274](#)
- unpack\_ims\_SLQSSIPCfgCallback\_ind\_t, [748](#)
  - ParamPresenceMask, [749](#)
  - SCTlv, [749](#)
  - SPTlv, [749](#)
  - SRTlv, [749](#)
  - STTlv, [749](#)
  - TT1Tlv, [749](#)
  - TT2Tlv, [749](#)
  - TTfTlv, [749](#)
- unpack\_ims\_SLQSSMSCfgCallback\_ind
  - ims.h, [1274](#)
- unpack\_ims\_SLQSSMSCfgCallback\_ind\_t, [749](#)
  - ParamPresenceMask, [750](#)
  - SFTlv, [750](#)
- unpack\_ims\_SLQSSetIMSSMSConfig
  - ims.h, [1272](#)
- unpack\_ims\_SLQSSetIMSSMSConfig\_t, [745](#)
  - pSettingResp, [746](#)
  - ParamPresenceMask, [746](#)
- unpack\_ims\_SLQSSetIMSUserConfig
  - ims.h, [1273](#)
- unpack\_ims\_SLQSSetIMSUserConfig\_t, [746](#)
  - pSettingResp, [746](#)
  - ParamPresenceMask, [746](#)
- unpack\_ims\_SLQSSetIMSVoIPConfig
  - ims.h, [1273](#)
- unpack\_ims\_SLQSSetIMSVoIPConfig\_t, [746](#)
  - pSettingResp, [747](#)
  - ParamPresenceMask, [747](#)
- unpack\_ims\_SLQSSetRegMgrConfig
  - ims.h, [1273](#)
- unpack\_ims\_SLQSSetRegMgrConfig\_t, [747](#)
  - pSettingResp, [747](#)
  - ParamPresenceMask, [747](#)
- unpack\_ims\_SLQSSetSIPConfig
  - ims.h, [1274](#)
- unpack\_ims\_SLQSSetSIPConfig\_t, [747](#)
  - pSettingResp, [748](#)
  - ParamPresenceMask, [748](#)
- unpack\_ims\_SLQSUserCfgCallback\_ind
  - ims.h, [1275](#)
- unpack\_ims\_SLQSUserCfgCallback\_ind\_t, [750](#)
  - IDTlv, [751](#)
  - ParamPresenceMask, [751](#)
- unpack\_ims\_SLQSVoIPCfgCallback\_ind
  - ims.h, [1275](#)
- unpack\_ims\_SLQSVoIPCfgCallback\_ind\_t, [751](#)
  - AMTlv, [752](#)
  - ParamPresenceMask, [753](#)
  - RTTlv, [753](#)
  - SDTlv, [753](#)
- unpack\_imsa\_SLQSGetIMSARegStatus
  - imsa.h, [1278](#)
- unpack\_imsa\_SLQSGetIMSARegStatus\_t, [753](#)
  - ImsRegErrCode, [754](#)
  - ImsRegStatus, [754](#)
  - NewImsRegStatus, [754](#)
  - ParamPresenceMask, [754](#)
- unpack\_imsa\_SLQSGetIMSAServiceStatus
  - imsa.h, [1279](#)
- unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, [754](#)
  - ParamPresenceMask, [756](#)
  - SmsServiceRat, [756](#)
  - SmsServiceStatus, [756](#)
  - UtServiceRat, [756](#)
  - UtServiceStatus, [756](#)
  - VoipServiceRat, [756](#)
  - VoipServiceStatus, [756](#)
  - VsServiceRat, [756](#)
  - VsServiceStatus, [756](#)
  - VtServiceRat, [756](#)
  - VtServiceStatus, [756](#)
- unpack\_imsa\_SLQSImsaPdpStatusCallback\_ind
  - imsa.h, [1279](#)
- unpack\_imsa\_SLQSImsaPdpStatusCallback\_ind\_t, [756](#)
  - FailErrCode, [757](#)
  - ParamPresenceMask, [757](#)
  - PdpConnState, [757](#)
- unpack\_imsa\_SLQSImsaRatStatusCallback\_ind
  - imsa.h, [1279](#)
- unpack\_imsa\_SLQSImsaRatStatusCallback\_ind\_t, [757](#)
  - ParamPresenceMask, [758](#)
  - RatHandover, [758](#)
- unpack\_imsa\_SLQSImsaRegStatusCallback\_ind
  - imsa.h, [1280](#)
- unpack\_imsa\_SLQSImsaRegStatusCallback\_ind\_t, [758](#)
  - IMSRegistration, [758](#)
  - ParamPresenceMask, [758](#)
- unpack\_imsa\_SLQSImsaSvcStatusCallback\_ind
  - imsa.h, [1280](#)
- unpack\_imsa\_SLQSImsaSvcStatusCallback\_ind\_t, [759](#)
  - ParamPresenceMask, [760](#)
  - SmsRat, [760](#)
  - SmsService, [760](#)
  - UtRat, [760](#)
  - UtService, [760](#)
  - VoipRat, [760](#)
  - VoipService, [760](#)
  - VtRat, [760](#)
  - VtService, [760](#)
- unpack\_imsa\_SLQSRegisterIMSAlndication
  - imsa.h, [1281](#)
- unpack\_imsa\_SLQSRegisterIMSAlndication\_t
  - imsa.h, [1277](#)
- unpack\_loc\_BestAvailPos\_Ind
  - loc.h, [1307](#)
- unpack\_loc\_BestAvailPos\_Ind\_t, [760](#)
  - pAltitudeWrtEllipsoid, [766](#)
  - pAltitudeWrtMeanSeaLevel, [766](#)
  - pGpsTime, [766](#)
  - pHeading, [766](#)
  - pHeadingUnc, [766](#)

- pHorCirConf, [766](#)
- pHorEllpConf, [766](#)
- pHorReliability, [766](#)
- pHorUncCircular, [766](#)
- pHorUncEllipseOrientAzimuth, [766](#)
- pHorUncEllipseSemiMajor, [766](#)
- pHorUncEllipseSemiMinor, [766](#)
- pLatitude, [766](#)
- pLongitude, [766](#)
- pMagneticDeviation, [766](#)
- pPrecisionDilution, [766](#)
- pSensorDataUsage, [766](#)
- pSpeedHorizontal, [766](#)
- pSpeedUnc, [766](#)
- pSpeedVertical, [766](#)
- pSpeedVerticalUnc, [766](#)
- pSvUsedforFix, [766](#)
- pTechnologyMask, [766](#)
- pTimeSrc, [766](#)
- pTimeUnc, [766](#)
- pTimestampUtc, [766](#)
- pVertConfidence, [766](#)
- pVertReliability, [767](#)
- pVertUnc, [767](#)
- pXid, [767](#)
- ParamPresenceMask, [766](#)
- status, [767](#)
- Tlvresult, [767](#)
- unpack\_loc\_CradleMountCallback\_Ind
  - loc.h, [1308](#)
- unpack\_loc\_CradleMountCallback\_Ind\_t, [767](#)
  - cradleMountConfigStatus, [767](#)
  - ParamPresenceMask, [767](#)
- unpack\_loc\_Delete\_Assist\_Data\_t, [767](#)
  - ParamPresenceMask, [768](#)
  - Tlvresult, [768](#)
- unpack\_loc\_DeleteAssistData
  - loc.h, [1308](#)
- unpack\_loc\_DeleteAssistData\_Ind
  - loc.h, [1308](#)
- unpack\_loc\_DeleteAssistData\_Ind\_t, [768](#)
  - ParamPresenceMask, [769](#)
  - status, [769](#)
  - Tlvresult, [769](#)
- unpack\_loc\_DeleteAssistData\_t
  - loc.h, [1301](#)
- unpack\_loc\_EngineState\_Ind
  - loc.h, [1309](#)
- unpack\_loc\_EngineState\_Ind\_t, [769](#)
  - engineState, [770](#)
  - ParamPresenceMask, [770](#)
  - Tlvresult, [770](#)
- unpack\_loc\_EventNMEA\_Ind
  - loc.h, [1309](#)
- unpack\_loc\_EventNMEA\_Ind\_t, [770](#)
  - NMEADData, [770](#)
  - ParamPresenceMask, [770](#)
  - Tlvresult, [770](#)
- unpack\_loc\_EventRegister
  - loc.h, [1309](#)
- unpack\_loc\_EventRegister\_t, [771](#)
  - ParamPresenceMask, [771](#)
  - Tlvresult, [771](#)
- unpack\_loc\_EventTimeSyncCallback\_Ind
  - loc.h, [1310](#)
- unpack\_loc\_EventTimeSyncCallback\_Ind\_t, [771](#)
  - ParamPresenceMask, [771](#)
  - timeSyncRefCount, [771](#)
- unpack\_loc\_FixCriteria\_Ind
  - loc.h, [1310](#)
- unpack\_loc\_FixCriteria\_Ind\_t, [771](#)
  - loc\_appProviderInfo, [772](#)
  - loc\_fixCriteriaStatus, [772](#)
  - loc\_horAccuracy, [772](#)
  - loc\_intermediateRptState, [772](#)
  - loc\_minInterval, [772](#)
  - ParamPresenceMask, [772](#)
- unpack\_loc\_GetFixCriteria
  - loc.h, [1310](#)
- unpack\_loc\_GetFixCriteria\_t
  - loc.h, [1301](#)
- unpack\_loc\_GetOpMode\_Ind
  - loc.h, [1311](#)
- unpack\_loc\_GetOpMode\_Ind\_t, [773](#)
  - pMode, [774](#)
  - ParamPresenceMask, [774](#)
  - Status, [774](#)
- unpack\_loc\_GetServer\_Ind
  - loc.h, [1311](#)
- unpack\_loc\_GetServer\_Ind\_t, [774](#)
  - pIPv4AddrInfo, [775](#)
  - pIPv6AddrInfo, [775](#)
  - pURL, [775](#)
  - ParamPresenceMask, [775](#)
  - serverStatus, [775](#)
  - serverType, [775](#)
  - Tlvresult, [775](#)
- unpack\_loc\_GnssSvInfo\_Ind
  - loc.h, [1311](#)
- unpack\_loc\_GnssSvInfo\_Ind\_t, [775](#)
  - altitudeAssumed, [776](#)
  - pSatelliteInfo, [776](#)
  - ParamPresenceMask, [776](#)
  - Tlvresult, [776](#)
- unpack\_loc\_InjectPositionCallback\_Ind
  - loc.h, [1312](#)
- unpack\_loc\_InjectPositionCallback\_Ind\_t, [776](#)
  - ParamPresenceMask, [777](#)
  - status, [777](#)
- unpack\_loc\_InjectSensorDataCallback\_Ind
  - loc.h, [1312](#)
- unpack\_loc\_InjectSensorDataCallback\_Ind\_t, [777](#)
  - injectSensorDataStatus, [778](#)
  - pAccelSamplesAccepted, [778](#)
  - pAccelTempSamplesAccepted, [778](#)
  - pGyroSamplesAccepted, [779](#)



- pGyroTempSamplesAccepted, 779
  - pOpaqueIdentifier, 779
  - ParamPresenceMask, 779
- unpack\_loc\_InjectTimeSyncDataCallback\_Ind
  - loc.h, 1312
- unpack\_loc\_InjectTimeSyncDataCallback\_Ind\_t, 779
  - injectTimeSyncStatus, 779
  - ParamPresenceMask, 779
- unpack\_loc\_InjectUTCTimeCallback\_Ind
  - loc.h, 1313
- unpack\_loc\_InjectUTCTimeCallback\_Ind\_t, 779
  - ParamPresenceMask, 780
  - status, 780
- unpack\_loc\_PositionRpt\_Ind
  - loc.h, 1313
- unpack\_loc\_PositionRpt\_Ind\_t, 780
  - pAltitudeAssumed, 786
  - pAltitudeWrtEllipsoid, 786
  - pAltitudeWrtMeanSeaLevel, 786
  - pFixId, 786
  - pGpsTime, 786
  - pHeading, 786
  - pHeadingUnc, 786
  - pHorConfidence, 786
  - pHorReliability, 786
  - pHorUncCircular, 786
  - pHorUncEllipseOrientAzimuth, 786
  - pHorUncEllipseSemiMajor, 786
  - pHorUncEllipseSemiMinor, 786
  - pLatitude, 786
  - pLeapSeconds, 786
  - pLongitude, 786
  - pMagneticDeviation, 786
  - pPrecisionDilution, 786
  - pSensorDataUsage, 786
  - pSpeedHorizontal, 786
  - pSpeedUnc, 786
  - pSpeedVertical, 786
  - pSvUsedforFix, 786
  - pTechnologyMask, 786
  - pTimeSrc, 786
  - pTimeUnc, 786
  - pTimestampUtc, 786
  - pVertConfidence, 787
  - pVertReliability, 787
  - pVertUnc, 787
  - ParamPresenceMask, 786
  - sessionId, 787
  - sessionStatus, 787
  - Tlvresult, 787
- unpack\_loc\_SLQSLOCGetBestAvailPos
  - loc.h, 1315
- unpack\_loc\_SLQSLOCGetBestAvailPos\_t, 791
  - ParamPresenceMask, 792
  - Tlvresult, 792
- unpack\_loc\_SLQSLOCGetOpMode
  - loc.h, 1316
- unpack\_loc\_SLQSLOCGetOpMode\_t, 792
  - ParamPresenceMask, 792
  - Tlvresult, 792
- unpack\_loc\_SLQSLOCGetServer
  - loc.h, 1316
- unpack\_loc\_SLQSLOCGetServer\_t
  - loc.h, 1301
- unpack\_loc\_SLQSLOCInjectPosition
  - loc.h, 1316
- unpack\_loc\_SLQSLOCInjectPosition\_t
  - loc.h, 1301
- unpack\_loc\_SLQSLOCInjectSensorData
  - loc.h, 1317
- unpack\_loc\_SLQSLOCInjectSensorData\_t
  - loc.h, 1301
- unpack\_loc\_SLQSLOCInjectUTCTime
  - loc.h, 1317
- unpack\_loc\_SLQSLOCInjectUTCTime\_t
  - loc.h, 1301
- unpack\_loc\_SLQSLOCSetCradleMountConfig
  - loc.h, 1317
- unpack\_loc\_SLQSLOCSetCradleMountConfig\_t
  - loc.h, 1301
- unpack\_loc\_SLQSLOCSetServer
  - loc.h, 1318
- unpack\_loc\_SensorStreamingCallback\_Ind
  - loc.h, 1313
- unpack\_loc\_SensorStreamingCallback\_Ind\_t, 787
  - pAccelAcceptReady, 787
  - pAccelTempAcceptReady, 787
  - pGyroAcceptReady, 788
  - pGyroTempAcceptReady, 788
  - ParamPresenceMask, 787
- unpack\_loc\_SetExtPowerConfig\_Ind
  - loc.h, 1314
- unpack\_loc\_SetExtPowerConfig\_Ind\_t, 788
  - ParamPresenceMask, 788
  - status, 788
  - Tlvresult, 788
- unpack\_loc\_SetExtPowerState
  - loc.h, 1314
- unpack\_loc\_SetExtPowerState\_t, 788
  - ParamPresenceMask, 789
  - Tlvresult, 789
- unpack\_loc\_SetOperationMode
  - loc.h, 1314
- unpack\_loc\_SetOperationMode\_Ind
  - loc.h, 1315
- unpack\_loc\_SetOperationMode\_Ind\_t, 789
  - ParamPresenceMask, 790
  - status, 790
  - Tlvresult, 790
- unpack\_loc\_SetOperationMode\_t, 790
  - ParamPresenceMask, 790
  - Tlvresult, 790
- unpack\_loc\_SetServer\_Ind
  - loc.h, 1315
- unpack\_loc\_SetServer\_Ind\_t, 790
  - ParamPresenceMask, 791

- serverStatus, [791](#)
- Tlvresult, [791](#)
- unpack\_loc\_Start
  - loc.h, [1318](#)
- unpack\_loc\_Start\_t, [792](#)
  - ParamPresenceMask, [793](#)
  - Tlvresult, [793](#)
- unpack\_loc\_Stop
  - loc.h, [1318](#)
- unpack\_loc\_Stop\_t, [793](#)
  - ParamPresenceMask, [793](#)
  - Tlvresult, [793](#)
- unpack\_nas\_GetACCOLC
  - nas.h, [1357](#)
- unpack\_nas\_GetACCOLC\_t, [793](#)
  - pAccolc, [794](#)
  - ParamPresenceMask, [794](#)
- unpack\_nas\_GetANAAAAAuthenticationStatus
  - nas.h, [1357](#)
- unpack\_nas\_GetANAAAAAuthenticationStatus\_t, [794](#)
  - pAuthStatus, [794](#)
  - ParamPresenceMask, [794](#)
- unpack\_nas\_GetCDMANetworkParameters
  - nas.h, [1358](#)
- unpack\_nas\_GetCDMANetworkParameters\_t, [794](#)
  - Application, [796](#)
  - Broadcast, [796](#)
  - CustomSCP, [796](#)
  - ForceRev0, [796](#)
  - ParamPresenceMask, [796](#)
  - Protocol, [796](#)
  - RegForeignNID, [796](#)
  - RegForeignSID, [796](#)
  - RegHomeSID, [797](#)
  - Roaming, [797](#)
  - SCI, [797](#)
  - SCM, [797](#)
- unpack\_nas\_GetHomeNetwork
  - nas.h, [1358](#)
- unpack\_nas\_GetHomeNetwork3GPP2
  - nas.h, [1358](#)
- unpack\_nas\_GetHomeNetwork3GPP2\_t, [797](#)
  - nameSize, [799](#)
  - pMCC, [799](#)
  - pMNC, [799](#)
  - pNID, [799](#)
  - pName, [799](#)
  - pNw2DescDisp, [799](#)
  - pNw2DescEnc, [799](#)
  - pNw2DescLen, [799](#)
  - pNw2MCC, [799](#)
  - pNw2MNC, [799](#)
  - pNw2Name, [799](#)
  - pSID, [799](#)
  - ParamPresenceMask, [799](#)
- unpack\_nas\_GetHomeNetwork\_t, [799](#)
  - mcc, [800](#)
  - mnc, [800](#)
  - name, [800](#)
  - nid, [800](#)
  - ParamPresenceMask, [800](#)
  - sid, [800](#)
- unpack\_nas\_GetNetworkPreference
  - nas.h, [1359](#)
- unpack\_nas\_GetNetworkPreference\_t, [800](#)
  - ActiveTechPref, [801](#)
  - Duration, [801](#)
  - ParamPresenceMask, [801](#)
  - PersistentTechPref, [801](#)
  - Tlvresult, [801](#)
- unpack\_nas\_GetRFInfo
  - nas.h, [1359](#)
- unpack\_nas\_GetRFInfo\_t, [801](#)
  - instancesSize, [802](#)
  - ParamPresenceMask, [802](#)
  - RFBandInfoElements, [802](#)
- unpack\_nas\_GetServingNetwork
  - nas.h, [1359](#)
- unpack\_nas\_GetServingNetwork\_t, [802](#)
  - CSDomain, [804](#)
  - DataCaps, [804](#)
  - DataCapsLen, [804](#)
  - MCC, [804](#)
  - MNC, [804](#)
  - Name, [804](#)
  - nameSize, [805](#)
  - PSDomain, [805](#)
  - ParamPresenceMask, [805](#)
  - RAN, [805](#)
  - Radiolfaces, [805](#)
  - RadiolfacesSize, [805](#)
  - RegistrationState, [805](#)
  - Roaming, [805](#)
- unpack\_nas\_GetServingNetworkCapabilities
  - nas.h, [1360](#)
- unpack\_nas\_GetServingNetworkCapabilities\_t, [805](#)
  - DataCaps, [806](#)
  - DataCapsLen, [806](#)
  - ParamPresenceMask, [806](#)
- unpack\_nas\_GetSignalStrengths
  - nas.h, [1360](#)
- unpack\_nas\_GetSignalStrengths\_t, [806](#)
  - len, [806](#)
  - ParamPresenceMask, [806](#)
  - radio, [806](#)
  - rsi, [806](#)
- unpack\_nas\_InitiateDomainAttach
  - nas.h, [1360](#)
- unpack\_nas\_InitiateDomainAttach\_t
  - nas.h, [1332](#)
- unpack\_nas\_PerformNetworkScan
  - nas.h, [1361](#)
- unpack\_nas\_PerformNetworkScan\_t, [807](#)
  - p3GppNetworkInfoInstances, [808](#)
  - p3GppNetworkInstanceSize, [808](#)
  - pLteOpModeTlv, [808](#)

- pPCIInfo, [808](#)
- pPCSInstance, [808](#)
- pPCSInstanceSize, [808](#)
- pRATInstance, [808](#)
- pRATInstanceSize, [808](#)
- pScanResult, [808](#)
- ParamPresenceMask, [808](#)
- unpack\_nas\_SLQSCfgSigInfo
  - nas.h, [1365](#)
- unpack\_nas\_SLQSCfgSigInfo\_t
  - nas.h, [1332](#)
- unpack\_nas\_SLQSGetErrorRate
  - nas.h, [1365](#)
- unpack\_nas\_SLQSGetErrorRate\_t, [813](#)
  - pCDMAFrameErrRate, [814](#)
  - pGSMBER, [814](#)
  - pHDRPackErrRate, [814](#)
  - pWCDMABER, [814](#)
  - ParamPresenceMask, [814](#)
- unpack\_nas\_SLQSGetHomeNetwork
  - nas.h, [1365](#)
- unpack\_nas\_SLQSGetHomeNetwork\_t, [814](#)
  - mcc, [815](#)
  - mnc, [816](#)
  - name, [816](#)
  - nid, [816](#)
  - pHomeNwMNC3Gpp, [816](#)
  - pNwNameSrc3Gpp, [816](#)
  - ParamPresenceMask, [816](#)
  - sid, [816](#)
- unpack\_nas\_SLQSGetNetworkTime
  - nas.h, [1366](#)
- unpack\_nas\_SLQSGetNetworkTime\_t, [816](#)
  - p3GPP2TimeInfo, [817](#)
  - p3GPPTIMEInfo, [817](#)
  - ParamPresenceMask, [817](#)
- unpack\_nas\_SLQSGetOperatorNameData
  - nas.h, [1366](#)
- unpack\_nas\_SLQSGetOperatorNameData\_t, [817](#)
  - pNITZInformation, [818](#)
  - pOperatorNameString, [818](#)
  - pOperatorPLMNList, [818](#)
  - pPLMNNetworkName, [818](#)
  - pSvcProviderName, [818](#)
  - ParamPresenceMask, [818](#)
- unpack\_nas\_SLQSGetPLMNName
  - nas.h, [1367](#)
- unpack\_nas\_SLQSGetPLMNName\_t, [818](#)
  - longName, [821](#)
  - longNameCI, [821](#)
  - longNameEn, [821](#)
  - longNameLen, [821](#)
  - longNameSB, [821](#)
  - ParamPresenceMask, [821](#)
  - shortName, [821](#)
  - shortNameCI, [821](#)
  - shortNameEn, [821](#)
  - shortNameLen, [821](#)
  - shortNameSB, [821](#)
  - spn, [821](#)
  - spnEncoding, [821](#)
  - spnLength, [822](#)
- unpack\_nas\_SLQSGetServingSystem
  - nas.h, [1367](#)
- unpack\_nas\_SLQSGetServingSystem\_t, [822](#)
  - BasestationID, [825](#)
  - BasestationLatitude, [825](#)
  - BasestationLongitude, [825](#)
  - CDMASystemInfoExt, [825](#)
  - CallBarStatus, [825](#)
  - CellID, [825](#)
  - ConcSvcInfo, [826](#)
  - CurrentPLMN, [826](#)
  - DTMInd, [826](#)
  - DataSrvCapabilities, [826](#)
  - DefaultRoamInd, [826](#)
  - DetailedSvcInfo, [826](#)
  - Gpp2TimeZone, [826](#)
  - GppNetworkDSTAdjustment, [826](#)
  - GppTimeZone, [826](#)
  - HdrPersonality, [826](#)
  - Lac, [826](#)
  - NetworkID, [826](#)
  - PRLInd, [826](#)
  - ParamPresenceMask, [826](#)
  - RoamIndicatorVal, [826](#)
  - RoamingIndicatorList, [826](#)
  - ServingSystem, [826](#)
  - SystemID, [826](#)
  - TrackAreaCode, [826](#)
- unpack\_nas\_SLQSGetServingSystemV2
  - nas.h, [1367](#)
- unpack\_nas\_SLQSGetServingSystemV2\_t, [826](#)
  - BasestationID, [830](#)
  - BasestationLatitude, [830](#)
  - BasestationLongitude, [830](#)
  - CDMASystemInfoExt, [830](#)
  - CallBarStatus, [830](#)
  - CellID, [830](#)
  - ConcSvcInfo, [830](#)
  - CurrentPLMN, [830](#)
  - DTMInd, [830](#)
  - DataSrvCapabilities, [830](#)
  - DefaultRoamInd, [830](#)
  - DetailedSvcInfo, [830](#)
  - Gpp2TimeZone, [830](#)
  - GppNetworkDSTAdjustment, [830](#)
  - GppTimeZone, [830](#)
  - HdrPersonality, [830](#)
  - Lac, [830](#)
  - NetworkID, [830](#)
  - PRLInd, [831](#)
  - ParamPresenceMask, [831](#)
  - RoamIndicatorVal, [831](#)
  - RoamingIndicatorList, [831](#)
  - ServingSystem, [831](#)

- SystemID, [831](#)
- TrackAreaCode, [831](#)
- unpack\_nas\_SLQSGetSignalStrength
  - nas.h, [1368](#)
- unpack\_nas\_SLQSGetSignalStrength\_t, [831](#)
  - ecioList, [833](#)
  - ecioListLen, [833](#)
  - errorRateList, [833](#)
  - errorRateListLen, [833](#)
  - lo, [833](#)
  - ltsr, [833](#)
  - ltsnr, [833](#)
  - ParamPresenceMask, [833](#)
  - rsrqInfo, [833](#)
  - rxSignalStrengthList, [833](#)
  - rxSignalStrengthListLen, [834](#)
  - signalStrengthReqMask, [834](#)
  - sinr, [834](#)
- unpack\_nas\_SLQSGetSysInfo
  - nas.h, [1368](#)
- unpack\_nas\_SLQSGetSysInfo\_t, [834](#)
  - pAddCDMASysInfo, [836](#)
  - pAddGSMSysInfo, [836](#)
  - pAddHDRSysInfo, [836](#)
  - pAddLTESysInfo, [836](#)
  - pCDMASrvStatusInfo, [837](#)
  - pCDMASysInfo, [837](#)
  - pCampedCiotLteOpMode, [837](#)
  - pGSMCallBarringSysInfo, [837](#)
  - pGSMCipherDomainSysInfo, [837](#)
  - pGSMSrvStatusInfo, [837](#)
  - pGSMSysInfo, [837](#)
  - pHDRSrvStatusInfo, [837](#)
  - pHDRSysInfo, [837](#)
  - pLTESrvStatusInfo, [837](#)
  - pLTESysInfo, [837](#)
  - pLTEVoiceSupportSysInfo, [837](#)
  - pWCDMASysInfo, [837](#)
  - ParamPresenceMask, [837](#)
- unpack\_nas\_SLQSGetSysInfoV2
  - nas.h, [1368](#)
- unpack\_nas\_SLQSGetSysInfoV2\_t, [837](#)
  - pAddGSMSysInfo, [841](#)
  - pAddHDRSysInfo, [841](#)
  - pAddLTESysInfo, [841](#)
  - pCDMASysInfo, [841](#)
  - pCampedCiotLteOpMode, [841](#)
  - pGSMSrvStatusInfo, [841](#)
  - pGSMSysInfo, [841](#)
  - pHDRSrvStatusInfo, [841](#)
  - pHDRSysInfo, [841](#)
  - plmsVoiceSupportLte, [841](#)
  - pLTESrvStatusInfo, [842](#)
  - pLTESysInfo, [842](#)
  - pLteEmbmsCoverage, [841](#)
  - pLteEmbmsTraceId, [842](#)
  - pLteRegDomain, [842](#)
  - pLteVoiceDomain, [842](#)
  - pNR5GCellStatusInfo, [842](#)
  - pNR5GSrvStatusInfo, [842](#)
  - pNR5GSysInfo, [842](#)
  - pSimRejInfo, [842](#)
  - pSrvRegRestriction, [842](#)
  - ParamPresenceMask, [841](#)
- unpack\_nas\_SLQSGetSysSelectionPref
  - nas.h, [1369](#)
- unpack\_nas\_SLQSGetSysSelectionPref\_t, [842](#)
  - pBandPref, [847](#)
  - pEmerMode, [847](#)
  - pGWAOrderPref, [847](#)
  - pLTEBandPref, [847](#)
  - pModePref, [847](#)
  - pNetSelPref, [847](#)
  - pPRLPref, [847](#)
  - pRoamPref, [847](#)
  - pSrvDomainPref, [847](#)
  - ParamPresenceMask, [847](#)
- unpack\_nas\_SLQSGetSysSelectionPrefExt
  - nas.h, [1369](#)
- unpack\_nas\_SLQSGetSysSelectionPrefExt\_t, [847](#)
  - pAcqOrderPref, [849](#)
  - pBandPref, [849](#)
  - pCiotAcqOrderPref, [849](#)
  - pCiotLteOpModePref, [849](#)
  - pEmerMode, [850](#)
  - pGWAOrderPref, [850](#)
  - pLTEBandPref, [850](#)
  - pLteM1BandPref, [850](#)
  - pLteNb1BandPref, [850](#)
  - pModePref, [850](#)
  - pNetSelPref, [850](#)
  - pNr5gBandPref, [850](#)
  - pPRLPref, [850](#)
  - pRatDisabledMask, [850](#)
  - pRoamPref, [850](#)
  - pSrvDomainPref, [850](#)
- unpack\_nas\_SLQSGetSysSelectionPrefExtV2
  - nas.h, [1370](#)
- unpack\_nas\_SLQSGetSysSelectionPrefExtV2\_t, [850](#)
  - pAcqOrderPref, [853](#)
  - pBandPref, [853](#)
  - pCiotAcqOrderPref, [853](#)
  - pEmerMode, [853](#)
  - pLteM1BandPref, [853](#)
  - pLteNb1BandPref, [853](#)
  - pModePref, [853](#)
  - pNetSelPref, [853](#)
  - pNr5gBandPref, [853](#)
  - pPRLPref, [853](#)
  - pRatDisabledMask, [853](#)
  - pRoamPref, [853](#)
  - pSrvDomainPref, [853](#)
  - pSrvRegRestrict, [853](#)
  - pUsageSetting, [854](#)
  - pVoiceDomainPref, [854](#)
  - ParamPresenceMask, [853](#)

- unpack\_nas\_SLQSNasInitiateNetworkRegistration  
nas.h, [1370](#)
- unpack\_nas\_SLQSNasInitiateNetworkRegistration\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasGetForbiddenNetworks  
nas.h, [1373](#)
- unpack\_nas\_SLQSNasGetForbiddenNetworks\_t, [862](#)  
ParamPresenceMask, [862](#)
- unpack\_nas\_SLQSNasGetDRXParams  
nas.h, [1372](#)
- unpack\_nas\_SLQSNasGetDRXParams\_t, [860](#)  
pCycleLen, [860](#)  
pEdrxEnable, [860](#)  
ParamPresenceMask, [860](#)
- unpack\_nas\_SLQSNasGetDRXParamsExt  
nas.h, [1373](#)
- unpack\_nas\_SLQSNasGetDRXParamsExt\_t, [861](#)
- unpack\_nas\_SLQSNasSetDRXParams  
nas.h, [1377](#)
- unpack\_nas\_SLQSNasSetDRXParams\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasSwtGetChannelLock  
nas.h, [1377](#)
- unpack\_nas\_SLQSNasSwtGetChannelLock\_t, [873](#)  
ParamPresenceMask, [873](#)
- unpack\_nas\_SLQSNasSwtSetChannelLock  
nas.h, [1379](#)
- unpack\_nas\_SLQSNasSwtSetChannelLock\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasConfigSigInfo2  
nas.h, [1370](#)
- unpack\_nas\_SLQSNasConfigSigInfo2\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind  
nas.h, [1371](#)
- unpack\_nas\_SLQSNasEdrxChangeInfoCallback\_Ind\_t,  
[854](#)
- unpack\_nas\_SLQSNasGet3GPP2Subscription  
nas.h, [1371](#)
- unpack\_nas\_SLQSNasGet3GPP2Subscription\_t, [855](#)  
pDirNum, [856](#)  
pMdn, [856](#)  
ParamPresenceMask, [856](#)
- unpack\_nas\_SLQSNasGetCellLocationInfo  
nas.h, [1371](#)
- unpack\_nas\_SLQSNasGetCellLocationInfo\_t, [856](#)  
pCDMAInfo, [857](#)  
pLTEInfoInterfreq, [857](#)  
pLTEInfoIntrafreq, [857](#)  
pUMTSInfo, [858](#)  
ParamPresenceMask, [857](#)
- unpack\_nas\_SLQSNasGetCellLocationInfoV2  
nas.h, [1372](#)
- unpack\_nas\_SLQSNasGetCellLocationInfoV2\_t, [858](#)  
pLteEarfcnInfo, [859](#)  
ParamPresenceMask, [859](#)
- unpack\_nas\_SLQSNasGetHDRColorCode  
nas.h, [1373](#)
- unpack\_nas\_SLQSNasGetHDRColorCode\_t, [863](#)  
pColorCode, [863](#)  
ParamPresenceMask, [863](#)
- unpack\_nas\_SLQSNasGetRFInfo  
nas.h, [1374](#)
- unpack\_nas\_SLQSNasGetRFInfo\_t, [863](#)  
pLTEOperationMode, [864](#)  
pRfBandInfoExtFormat, [864](#)  
pRfBandwidthInfo, [864](#)  
pRfDedicatedBandInfo, [864](#)  
rfbandInfoList, [864](#)  
Tlvresult, [864](#)
- unpack\_nas\_SLQSNasGetSigInfo  
nas.h, [1374](#)
- unpack\_nas\_SLQSNasGetSigInfo\_t, [864](#)  
CDMASSInfo, [865](#)  
GSMSSInfo, [865](#)  
HDRSSInfo, [865](#)  
LTESSInfo, [865](#)  
ParamPresenceMask, [865](#)
- unpack\_nas\_SLQSNasGetTxRxInfo  
nas.h, [1374](#)
- unpack\_nas\_SLQSNasGetTxRxInfo\_t, [865](#)  
pRXChain0Info, [866](#)  
pRXChain1Info, [866](#)  
pTXInfo, [866](#)  
ParamPresenceMask, [866](#)
- unpack\_nas\_SLQSNasIndicationRegisterExt  
nas.h, [1375](#)
- unpack\_nas\_SLQSNasIndicationRegisterExt\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasIndicationRegisterV2  
nas.h, [1375](#)
- unpack\_nas\_SLQSNasIndicationRegisterV2\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind  
nas.h, [1375](#)
- unpack\_nas\_SLQSNasNetworkRejectCallback\_Ind\_t,  
[866](#)  
pCsgId, [869](#)  
pLteOpMode, [869](#)  
pPlmnId, [869](#)  
ParamPresenceMask, [869](#)  
radioIlf, [869](#)  
rejectCause, [869](#)  
serviceDomain, [869](#)
- unpack\_nas\_SLQSNasNetworkTimeCallback\_ind  
nas.h, [1376](#)
- unpack\_nas\_SLQSNasNetworkTimeCallback\_ind\_t,  
[869](#)  
pDayltSavAdj, [870](#)  
pRadioInterface, [870](#)  
pTimeZone, [870](#)  
ParamPresenceMask, [870](#)  
universalTime, [870](#)
- unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind  
nas.h, [1376](#)
- unpack\_nas\_SLQSNasRFBandInfoCallback\_Ind\_t, [870](#)

- unpack\_nas\_SLQSNasSigInfoCallback\_ind  
nas.h, [1377](#)
- unpack\_nas\_SLQSNasSigInfoCallback\_ind\_t, [871](#)  
pRscp, [873](#)  
ParamPresenceMask, [872](#)
- unpack\_nas\_SLQSNasSwiIndicationRegister  
nas.h, [1378](#)
- unpack\_nas\_SLQSNasSwiIndicationRegister\_t  
nas.h, [1332](#)
- unpack\_nas\_SLQSNasSwiModemStatus  
nas.h, [1378](#)
- unpack\_nas\_SLQSNasSwiModemStatus\_t, [874](#)  
commonInfo, [874](#)  
pLTEInfo, [874](#)  
ParamPresenceMask, [874](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind  
nas.h, [1378](#)
- unpack\_nas\_SLQSNasSwiOTAMessageCallback\_ind\_t, [874](#)  
Info, [875](#)  
Tlvresult, [875](#)
- unpack\_nas\_SLQSNasSysInfoCallback\_ind  
nas.h, [1379](#)
- unpack\_nas\_SLQSNasTimerCallback\_ind  
nas.h, [1379](#)
- unpack\_nas\_SLQSNasTimerCallback\_ind\_t, [875](#)  
ParamPresenceMask, [875](#)  
t3396\_apn, [875](#)  
t3396\_plmn\_id, [876](#)  
t3396\_val, [876](#)
- unpack\_nas\_SLQSPerformNetworkScanV2  
nas.h, [1380](#)
- unpack\_nas\_SLQSPerformNetworkScanV2\_t, [876](#)  
p3GppNetworkInfoInstances, [877](#)  
p3GppNetworkInstanceSize, [877](#)  
pLteOpModeTlv, [877](#)  
pNetworkNameSrcTlv, [877](#)  
pPCIInfo, [877](#)  
pPcsInstance, [877](#)  
pPcsInstanceSize, [877](#)  
pRATInstance, [877](#)  
pRATInstanceSize, [877](#)  
pScanResult, [877](#)  
ParamPresenceMask, [877](#)
- unpack\_nas\_SLQSSetBandPreference  
nas.h, [1380](#)
- unpack\_nas\_SLQSSetBandPreference\_t  
nas.h, [1333](#)
- unpack\_nas\_SLQSSetSignalStrengthsCallback  
nas.h, [1380](#)
- unpack\_nas\_SLQSSetSignalStrengthsCallback\_t  
nas.h, [1333](#)
- unpack\_nas\_SLQSSetSysSelectionPref  
nas.h, [1381](#)
- unpack\_nas\_SLQSSetSysSelectionPref\_t  
nas.h, [1333](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind  
nas.h, [1381](#)
- unpack\_nas\_SLQSSetSysSelectionPrefCallBack\_ind\_t,  
[878](#)  
Info, [878](#)  
Tlvresult, [878](#)
- unpack\_nas\_SLQSSetSysSelectionPrefExt  
nas.h, [1381](#)
- unpack\_nas\_SLQSSetSysSelectionPrefExt\_t  
nas.h, [1333](#)
- unpack\_nas\_SLQSSwiGetHDRPersonality  
nas.h, [1382](#)
- unpack\_nas\_SLQSSwiGetHDRPersonality\_t, [878](#)  
pCurrentPersonality, [879](#)  
pPersonalityListLength, [879](#)  
pProtocolSubtypeElement, [879](#)  
ParamPresenceMask, [879](#)
- unpack\_nas\_SLQSSwiGetHDRProtSubtype  
nas.h, [1382](#)
- unpack\_nas\_SLQSSwiGetHDRProtSubtype\_t, [879](#)  
pAppSubType, [880](#)  
pCurrentPrsnlty, [880](#)  
pPersonalityListLength, [880](#)  
ParamPresenceMask, [880](#)
- unpack\_nas\_SLQSSwiGetHRPDStats  
nas.h, [1382](#)
- unpack\_nas\_SLQSSwiGetHRPDStats\_t, [880](#)  
pPilotSetData, [881](#)  
pUATI, [881](#)  
ParamPresenceMask, [881](#)
- unpack\_nas\_SLQSSwiGetLteCQI  
nas.h, [1383](#)
- unpack\_nas\_SLQSSwiGetLteCQI\_t, [881](#)  
ParamPresenceMask, [882](#)  
ValidityCW0, [882](#)  
ValidityCW1, [882](#)
- unpack\_nas\_SLQSSwiGetLteSccRxInfo  
nas.h, [1383](#)
- unpack\_nas\_SLQSSwiGetLteSccRxInfo\_t, [882](#)  
pSccRxInfo, [882](#)  
ParamPresenceMask, [882](#)
- unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind  
nas.h, [1383](#)
- unpack\_nas\_SLQSSwiHDRPersonalityCallback\_Ind\_t,  
[882](#)
- unpack\_nas\_SLQSSwiNetworkDebug  
nas.h, [1384](#)
- unpack\_nas\_SLQSSwiNetworkDebug\_t, [883](#)  
pDataStatusDetail, [884](#)  
pDeviceConfigDetail, [884](#)  
pNetworkStat1x, [884](#)  
pNetworkStatEVDO, [884](#)  
pObjectVer, [884](#)  
ParamPresenceMask, [884](#)
- unpack\_nas\_SLQSSwiPSDetach  
nas.h, [1384](#)
- unpack\_nas\_SLQSSwiPSDetach\_t  
nas.h, [1333](#)
- unpack\_nas\_SLQSSwiRandIndicatorCallback\_Ind  
nas.h, [1384](#)



- unpack\_nas\_SLQSSwiRandIndicatorCallback\_Ind\_t, 884
  - ParamPresenceMask, 884
  - rankIndicatorTlv, 884
- unpack\_nas\_SLQSSysInfoCallback\_ind\_t, 884
  - pGSMSSysInfo, 888
  - pHDRSysInfo, 888
  - pLTESysInfo, 888
  - pLteCiotOpModeTlv, 888
  - pNR5GCellStatus, 888
  - pSysInfoNoChange, 888
  - ParamPresenceMask, 888
- unpack\_nas\_SetACCOLC
  - nas.h, 1361
- unpack\_nas\_SetACCOLC\_t
  - nas.h, 1332
- unpack\_nas\_SetCDMANetworkParameters
  - nas.h, 1361
- unpack\_nas\_SetCDMANetworkParameters\_t
  - nas.h, 1332
- unpack\_nas\_SetDataCapabilitiesCallback\_ind
  - nas.h, 1362
- unpack\_nas\_SetDataCapabilitiesCallback\_ind\_t, 808
  - dataCaps, 809
  - dataCapsSize, 809
  - ParamPresenceMask, 809
- unpack\_nas\_SetEventReportInd
  - nas.h, 1362
- unpack\_nas\_SetEventReportInd\_t, 809
  - ParamPresenceMask, 810
  - RFTlv, 810
  - RRTlv, 810
  - SLQSSSTlv, 810
  - SSTlv, 810
- unpack\_nas\_SetLURejectCallback
  - nas.h, 1363
- unpack\_nas\_SetLURejectCallback\_t
  - nas.h, 1332
- unpack\_nas\_SetNasLTECphyCaIndCallback\_ind
  - nas.h, 1363
- unpack\_nas\_SetNasLTECphyCaIndCallback\_ind\_t, 810
  - ParamPresenceMask, 811
  - sNumScellsConfig, 811
- unpack\_nas\_SetNetworkPreference
  - nas.h, 1363
- unpack\_nas\_SetNetworkPreference\_t, 811
  - ParamPresenceMask, 812
  - Tlvresult, 812
- unpack\_nas\_SetRFInfoCallback
  - nas.h, 1364
- unpack\_nas\_SetRFInfoCallback\_t
  - nas.h, 1332
- unpack\_nas\_SetRoamingIndicatorCallback\_ind
  - nas.h, 1364
- unpack\_nas\_SetRoamingIndicatorCallback\_ind\_t, 812
  - ParamPresenceMask, 812
  - roaming, 812
- unpack\_nas\_SetServingSystemCallback\_ind
  - nas.h, 1364
- unpack\_nas\_SetServingSystemCallback\_ind\_t, 812
  - ParamPresenceMask, 813
  - SSInfo, 813
  - Tlvresult, 813
- unpack\_nas\_SlqsGetLTECphyCAInfo
  - nas.h, 1366
- unpack\_nas\_SlqsGetLTECphyCAInfo\_t, 816
  - LTECphyCAInfo, 816
  - ParamPresenceMask, 816
  - Tlvresult, 816
- unpack\_omaDmConfigTlv\_t, 888
  - alertmsg, 889
  - alertmsglength, 889
  - state, 889
  - userInputReq, 889
  - userInputTimeout, 889
- unpack\_omaDmFotaTlv\_t, 889
  - description, 891
  - descriptionlength, 891
  - fwdloadsize, 891
  - fwloadComplete, 891
  - namelength, 891
  - package\_name, 891
  - sessionType, 891
  - severity, 891
  - state, 891
  - updateCompleteStatus, 891
  - userInputReq, 891
  - userInputTimeout, 891
  - version, 891
  - versionlength, 891
- unpack\_omaDmNotificationsTlv\_t, 891
  - notification, 892
  - sessionStatus, 892
- unpack\_pds\_ForceXTRADownload
  - pds.h, 1400
- unpack\_pds\_ForceXTRADownload\_t
  - pds.h, 1390
- unpack\_pds\_GetPDSDDefaults
  - pds.h, 1400
- unpack\_pds\_GetPDSDDefaults\_t, 892
  - pAccuracy, 893
  - pInterval, 893
  - pOperation, 893
  - pTimeout, 893
  - ParamPresenceMask, 893
- unpack\_pds\_GetPDSSState
  - pds.h, 1400
- unpack\_pds\_GetPDSSState\_t, 893
  - pEnabledStatus, 894
  - pTrackingStatus, 894
  - ParamPresenceMask, 894
- unpack\_pds\_GetPortAutomaticTracking
  - pds.h, 1401
- unpack\_pds\_GetPortAutomaticTracking\_t, 894
  - ParamPresenceMask, 894
  - pbAuto, 895

- unpack\_pds\_GetServiceAutomaticTracking
  - pds.h, [1401](#)
- unpack\_pds\_GetServiceAutomaticTracking\_t, [895](#)
  - ParamPresenceMask, [895](#)
  - pbAuto, [895](#)
- unpack\_pds\_GetXTRAAutomaticDownload
  - pds.h, [1401](#)
- unpack\_pds\_GetXTRAAutomaticDownload\_t, [895](#)
  - pInterval, [896](#)
  - ParamPresenceMask, [896](#)
  - pbEnabled, [896](#)
- unpack\_pds\_GetXTRANetwork
  - pds.h, [1402](#)
- unpack\_pds\_GetXTRANetwork\_t, [896](#)
  - pPreference, [896](#)
  - ParamPresenceMask, [896](#)
- unpack\_pds\_GetXTRAValidity
  - pds.h, [1402](#)
- unpack\_pds\_GetXTRAValidity\_t, [896](#)
  - pDuration, [897](#)
  - pGPSWeek, [897](#)
  - pGPSWeekOffset, [897](#)
  - ParamPresenceMask, [897](#)
- unpack\_pds\_PDSInjectTimeReference
  - pds.h, [1403](#)
- unpack\_pds\_PDSInjectTimeReference\_t
  - pds.h, [1390](#)
- unpack\_pds\_ResetPDSDData
  - pds.h, [1403](#)
- unpack\_pds\_ResetPDSDData\_t
  - pds.h, [1390](#)
- unpack\_pds\_SLQSGetAGPSConfig
  - pds.h, [1406](#)
- unpack\_pds\_SLQSGetAGPSConfig\_t, [900](#)
  - pServerAddress, [901](#)
  - pServerPort, [901](#)
  - pServerURL, [901](#)
  - ParamPresenceMask, [901](#)
- unpack\_pds\_SLQSGetGPSStateInfo
  - pds.h, [1407](#)
- unpack\_pds\_SLQSGetGPSStateInfo\_t, [901](#)
  - Altitude, [905](#)
  - EngineState, [905](#)
  - HorizontalUncertainty, [905](#)
  - lono\_valid, [905](#)
  - Latitude, [905](#)
  - Longitude, [905](#)
  - ParamPresenceMask, [905](#)
  - Time\_uncert\_ms, [905](#)
  - ValidMask, [905](#)
  - VerticalUncertainty, [905](#)
- unpack\_pds\_SLQSPDSDeterminePosition
  - pds.h, [1407](#)
- unpack\_pds\_SLQSPDSDeterminePosition\_t
  - pds.h, [1390](#)
- unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference
  - pds.h, [1407](#)
- unpack\_pds\_SLQSPDSInjectAbsoluteTimeReference\_t
  - pds.h, [1390](#)
- unpack\_pds\_SLQSPDSInjectPositionData
  - pds.h, [1408](#)
- unpack\_pds\_SLQSPDSInjectPositionData\_t
  - pds.h, [1390](#)
- unpack\_pds\_SLQSSetAGPSConfig
  - pds.h, [1408](#)
- unpack\_pds\_SLQSSetAGPSConfig\_t
  - pds.h, [1390](#)
- unpack\_pds\_SLQSSetPositionMethodState
  - pds.h, [1408](#)
- unpack\_pds\_SLQSSetPositionMethodState\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetEventReport\_Ind
  - pds.h, [1403](#)
- unpack\_pds\_SetEventReport\_Ind\_t, [897](#)
  - dLatitude, [899](#)
  - dLongitude, [899](#)
  - has\_PositionDataNMEA, [899](#)
  - has\_SessionStatus, [899](#)
  - has\_dLatitude, [899](#)
  - has\_dLongitude, [899](#)
  - has\_posSrc, [899](#)
  - ParamPresenceMask, [899](#)
  - posSrc, [899](#)
  - PositionDataNMEA, [899](#)
  - SessionStatus, [899](#)
- unpack\_pds\_SetEventReportCallback
  - pds.h, [1404](#)
- unpack\_pds\_SetEventReportCallback\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetPDSDDefaults
  - pds.h, [1404](#)
- unpack\_pds\_SetPDSDDefaults\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetPDSSState
  - pds.h, [1404](#)
- unpack\_pds\_SetPDSSState\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetPdsState\_Ind
  - pds.h, [1405](#)
- unpack\_pds\_SetPdsState\_Ind\_t, [899](#)
  - EnabledStatus, [900](#)
  - ParamPresenceMask, [900](#)
  - TlvPresent, [900](#)
  - TrackingStatus, [900](#)
- unpack\_pds\_SetPortAutomaticTracking
  - pds.h, [1405](#)
- unpack\_pds\_SetPortAutomaticTracking\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetServiceAutomaticTracking
  - pds.h, [1405](#)
- unpack\_pds\_SetServiceAutomaticTracking\_t
  - pds.h, [1390](#)
- unpack\_pds\_SetXTRAAutomaticDownload
  - pds.h, [1406](#)
- unpack\_pds\_SetXTRAAutomaticDownload\_t
  - pds.h, [1390](#)



- unpack\_pds\_SetXTRANetwork
  - pds.h, [1406](#)
- unpack\_pds\_SetXTRANetwork\_t
  - pds.h, [1390](#)
- unpack\_pds\_StartPDSTrackingSessionExt
  - pds.h, [1409](#)
- unpack\_pds\_StartPDSTrackingSessionExt\_t
  - pds.h, [1390](#)
- unpack\_pds\_StopPDSTrackingSession
  - pds.h, [1409](#)
- unpack\_pds\_StopPDSTrackingSession\_t
  - pds.h, [1390](#)
- unpack\_qmi\_t, [906](#)
  - msgid, [906](#)
  - type, [906](#)
  - xid, [906](#)
- unpack\_qos\_BindDataPort
  - qos.h, [1461](#)
- unpack\_qos\_BindDataPort\_t, [906](#)
  - ParamPresenceMask, [906](#)
- unpack\_qos\_IPv4Addr\_t, [907](#)
  - addr, [907](#)
  - subnetMask, [907](#)
- unpack\_qos\_IPv6Addr\_t, [907](#)
  - addr, [908](#)
  - prefixLen, [908](#)
- unpack\_qos\_IPv6TrafCls\_t, [908](#)
  - mask, [908](#)
  - val, [908](#)
- unpack\_qos\_Port\_t, [909](#)
  - port, [909](#)
  - range, [909](#)
- unpack\_qos\_QosFlowInfo\_t, [909](#)
  - BearerID, [910](#)
  - is\_RxQFlowGranted\_Available, [910](#)
  - is\_TxQFlowGranted\_Available, [910](#)
  - NumRxFilters, [910](#)
  - NumTxFilters, [911](#)
  - QFlowState, [911](#)
  - RxQFilter, [911](#)
  - RxQFlowGranted, [911](#)
  - TxQFilter, [911](#)
  - TxQFlowGranted, [911](#)
- unpack\_qos\_QosFlowInfoState\_t, [911](#)
  - id, [911](#)
  - isNewFlow, [911](#)
  - state, [911](#)
- unpack\_qos\_SLQSQosGetNetworkStatus
  - qos.h, [1461](#)
- unpack\_qos\_SLQSQosGetNetworkStatus\_t, [911](#)
  - NWQoSStatus, [912](#)
  - ParamPresenceMask, [912](#)
- unpack\_qos\_SLQSQosSwiReadApnExtraParams
  - qos.h, [1462](#)
- unpack\_qos\_SLQSQosSwiReadApnExtraParams\_t, [912](#)
  - ambr\_dl, [913](#)
  - ambr\_dl\_ext, [913](#)
  - ambr\_dl\_ext2, [913](#)
  - ambr\_ul, [913](#)
  - ambr\_ul\_ext, [913](#)
  - ambr\_ul\_ext2, [913](#)
  - apnId, [913](#)
  - ParamPresenceMask, [913](#)
- unpack\_qos\_SLQSQosSwiReadDataStats
  - qos.h, [1462](#)
- unpack\_qos\_SLQSQosSwiReadDataStats\_t, [914](#)
  - apnId, [915](#)
  - numQosFlow, [915](#)
  - ParamPresenceMask, [915](#)
  - qosFlow, [915](#)
  - total\_rx\_bytes, [915](#)
  - total\_rx\_pkt, [915](#)
  - total\_tx\_bytes, [915](#)
  - total\_tx\_pkt, [915](#)
- unpack\_qos\_SLQSSetQosEventCallback
  - qos.h, [1463](#)
- unpack\_qos\_SLQSSetQosEventCallback\_ind
  - qos.h, [1463](#)
- unpack\_qos\_SLQSSetQosEventCallback\_ind\_t, [915](#)
  - NumFlows, [916](#)
  - ParamPresenceMask, [916](#)
  - QosFlowInfo, [916](#)
- unpack\_qos\_SLQSSetQosEventCallback\_t, [916](#)
  - ParamPresenceMask, [916](#)
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind
  - qos.h, [1464](#)
- unpack\_qos\_SLQSSetQosNWStatusCallback\_ind\_t, [916](#)
  - status, [917](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind
  - qos.h, [1464](#)
- unpack\_qos\_SLQSSetQosPriEventCallback\_ind\_t, [917](#)
  - event, [917](#)
  - ParamPresenceMask, [917](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind
  - qos.h, [1465](#)
- unpack\_qos\_SLQSSetQosStatusCallback\_ind\_t, [917](#)
  - event, [919](#)
  - id, [919](#)
  - ParamPresenceMask, [919](#)
  - reason, [919](#)
  - status, [919](#)
- unpack\_qos\_Tos\_t, [927](#)
  - mask, [927](#)
  - val, [927](#)
- unpack\_qos\_dataRate\_t, [907](#)
  - dataRateMax, [907](#)
  - guaranteedRate, [907](#)
- unpack\_qos\_pktErrRate\_t, [908](#)
  - exponent, [909](#)
  - multiplier, [909](#)
- unpack\_qos\_swiQosFilter\_t, [919](#)
  - EspSpi, [921](#)
  - IPv4DstAddr, [921](#)
  - IPv4SrcAddr, [921](#)

- IPv4Tos, [921](#)
- IPv6DstAddr, [921](#)
- IPv6Label, [921](#)
- IPv6SrcAddr, [921](#)
- IPv6TrafCls, [921](#)
- Id, [921](#)
- index, [921](#)
- is\_EspSpi\_Available, [921](#)
- is\_IPv4DstAddr\_Available, [921](#)
- is\_IPv4SrcAddr\_Available, [921](#)
- is\_IPv4Tos\_Available, [921](#)
- is\_IPv6DstAddr\_Available, [921](#)
- is\_IPv6Label\_Available, [921](#)
- is\_IPv6SrcAddr\_Available, [921](#)
- is\_IPv6TrafCls\_Available, [921](#)
- is\_Id\_Available, [921](#)
- is\_NxtHdrProto\_Available, [922](#)
- is\_Precedence\_Available, [922](#)
- is\_TCPDstPort\_Available, [922](#)
- is\_TCPSrcPort\_Available, [922](#)
- is\_TranDstPort\_Available, [922](#)
- is\_TranSrcPort\_Available, [922](#)
- is\_UDPDstPort\_Available, [922](#)
- is\_UDPSrcPort\_Available, [922](#)
- NxtHdrProto, [922](#)
- Precedence, [922](#)
- TCPDstPort, [922](#)
- TCPSrcPort, [922](#)
- TranDstPort, [922](#)
- TranSrcPort, [922](#)
- UDPDstPort, [922](#)
- UDPSrcPort, [922](#)
- version, [922](#)
- unpack\_qos\_swiQosFlow\_t, [922](#)
  - DataRate, [925](#)
  - index, [925](#)
  - is\_DataRate\_Available, [925](#)
  - is\_Jitter\_Available, [925](#)
  - is\_Latency\_Available, [925](#)
  - is\_LteQci\_Available, [925](#)
  - is\_MaxAllowedPktSz\_Available, [925](#)
  - is\_MinPolicedPktSz\_Available, [925](#)
  - is\_PktErrRate\_Available, [925](#)
  - is\_ProfileId3GPP2\_Available, [925](#)
  - is\_TokenBucket\_Available, [925](#)
  - is\_TrafficClass\_Available, [925](#)
  - is\_val\_3GPP2Pri\_Available, [925](#)
  - is\_val\_3GPPImCn\_Available, [925](#)
  - is\_val\_3GPPSigInd\_Available, [926](#)
  - Jitter, [926](#)
  - Latency, [926](#)
  - LteQci, [926](#)
  - MaxAllowedPktSz, [926](#)
  - MinPolicedPktSz, [926](#)
  - PktErrRate, [926](#)
  - ProfileId3GPP2, [926](#)
  - TokenBucket, [926](#)
  - TrafficClass, [926](#)
  - val\_3GPP2Pri, [926](#)
  - val\_3GPPImCn, [926](#)
  - val\_3GPPResResidualBER, [926](#)
  - val\_3GPPSigInd, [926](#)
  - val\_3GPPTraHdlPri, [926](#)
- unpack\_qos\_tokenBucket\_t, [926](#)
  - bucketSz, [927](#)
  - peakRate, [927](#)
  - tokenRate, [927](#)
- unpack\_result\_code\_only
  - common.h, [1198](#)
- unpack\_result\_t, [928](#)
  - ParamPresenceMask, [928](#)
  - Tlvresult, [928](#)
- unpack\_rms\_GetSMSWake
  - rms.h, [1467](#)
- unpack\_rms\_GetSMSWake\_t, [929](#)
  - enabled, [929](#)
  - ParamPresenceMask, [929](#)
  - wake\_mask, [929](#)
- unpack\_rms\_SetSMSWake
  - rms.h, [1467](#)
- unpack\_rms\_SetSMSWake\_t, [929](#)
  - ParamPresenceMask, [930](#)
- unpack\_sar\_SLQSGetRfSarState
  - sar.h, [1469](#)
- unpack\_sar\_SLQSGetRfSarState\_t, [931](#)
  - pSarRFState, [931](#)
  - ParamPresenceMask, [931](#)
- unpack\_sar\_SLQSSetRfSarState
  - sar.h, [1469](#)
- unpack\_sar\_SLQSSetRfSarState\_t
  - sar.h, [1468](#)
- unpack\_sms\_GetSMSCAddress
  - sms.h, [1482](#)
- unpack\_sms\_GetSMSCAddress\_t, [931](#)
  - addressSize, [932](#)
  - pSMSCAddress, [932](#)
  - pSMSCType, [932](#)
  - ParamPresenceMask, [932](#)
  - typeSize, [932](#)
- unpack\_sms\_SLQSDeleteSMS
  - sms.h, [1484](#)
- unpack\_sms\_SLQSDeleteSMS\_t, [936](#)
  - ParamPresenceMask, [936](#)
- unpack\_sms\_SLQSGetIndicationRegister
  - sms.h, [1484](#)
- unpack\_sms\_SLQSGetIndicationRegister\_t, [936](#)
  - pGetIndicationRegInfo, [937](#)
  - ParamPresenceMask, [937](#)
- unpack\_sms\_SLQSGetMessageWaiting
  - sms.h, [1485](#)
- unpack\_sms\_SLQSGetMessageWaiting\_t, [937](#)
  - pGetMsgWaitingInfoResp, [937](#)
  - ParamPresenceMask, [937](#)
- unpack\_sms\_SLQSGetSMS
  - sms.h, [1485](#)
- unpack\_sms\_SLQSGetSMS\_t, [937](#)

- message, 938
- messageFormat, 938
- messageSize, 938
- messageTag, 938
- ParamPresenceMask, 938
- unpack\_sms\_SLQSGetSMSList
  - sms.h, 1486
- unpack\_sms\_SLQSGetSMSList\_t, 939
  - messageList, 940
  - messageListSize, 940
  - ParamPresenceMask, 940
- unpack\_sms\_SLQSGetSmsBroadcastConfig
  - sms.h, 1485
- unpack\_sms\_SLQSGetSmsBroadcastConfig\_t, 938
  - pBroadcastConfig, 939
  - pCDMABroadcastConfig, 939
  - ParamPresenceMask, 939
- unpack\_sms\_SLQSGetTransLayerInfo
  - sms.h, 1486
- unpack\_sms\_SLQSGetTransLayerInfo\_t, 940
  - pGetTransLayerInfo, 940
  - ParamPresenceMask, 940
- unpack\_sms\_SLQSGetTransNWRegInfo
  - sms.h, 1487
- unpack\_sms\_SLQSGetTransNWRegInfo\_t, 940
  - ParamPresenceMask, 941
- unpack\_sms\_SLQSModifySMSStatus
  - sms.h, 1487
- unpack\_sms\_SLQSModifySMSStatus\_t, 941
  - ParamPresenceMask, 941
- unpack\_sms\_SLQSNWRegInfoCallback\_ind
  - sms.h, 1487
- unpack\_sms\_SLQSNWRegInfoCallback\_ind\_t, 941
  - NWRegStat, 942
  - ParamPresenceMask, 942
- unpack\_sms\_SLQSSendAsyncSMS
  - sms.h, 1488
- unpack\_sms\_SLQSSendAsyncSMS\_t, 942
  - ParamPresenceMask, 942
- unpack\_sms\_SLQSSetIndicationRegister
  - sms.h, 1488
- unpack\_sms\_SLQSSetIndicationRegister\_t, 942
  - ParamPresenceMask, 943
- unpack\_sms\_SLQSSetSmsBroadcastActivation
  - sms.h, 1488
- unpack\_sms\_SLQSSetSmsBroadcastActivation\_t, 943
  - ParamPresenceMask, 943
- unpack\_sms\_SLQSSetSmsBroadcastConfig
  - sms.h, 1489
- unpack\_sms\_SLQSSetSmsBroadcastConfig\_t, 943
  - ParamPresenceMask, 944
- unpack\_sms\_SLQSSetSmsStorage
  - sms.h, 1489
- unpack\_sms\_SLQSSetSmsStorage\_t, 944
  - ParamPresenceMask, 944
- unpack\_sms\_SLQSSmsGetMaxStorageSize
  - sms.h, 1489
- unpack\_sms\_SLQSSmsGetMaxStorageSize\_t, 944
  - pMaxStorageSizeResp, 945
  - ParamPresenceMask, 944
- unpack\_sms\_SLQSSmsGetMessageProtocol
  - sms.h, 1490
- unpack\_sms\_SLQSSmsGetMessageProtocol\_t, 945
  - pMessageProtocol, 945
  - ParamPresenceMask, 945
- unpack\_sms\_SLQSSmsSetRoutes
  - sms.h, 1490
- unpack\_sms\_SLQSSmsSetRoutes\_t, 945
  - ParamPresenceMask, 945
- unpack\_sms\_SLQSSwiGetSMSStorage
  - sms.h, 1490
- unpack\_sms\_SLQSSwiGetSMSStorage\_t, 946
  - pSmsStorage, 946
  - ParamPresenceMask, 946
- unpack\_sms\_SLQSTransLayerInfoCallback\_ind
  - sms.h, 1491
- unpack\_sms\_SLQSTransLayerInfoCallback\_ind\_t, 946
  - pTransLayerInfo, 947
  - ParamPresenceMask, 947
  - regInd, 947
- unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind
  - sms.h, 1491
- unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t, 947
  - causeCode, 949
  - errorClass, 949
  - messageID, 949
  - RPCause, 949
  - sendStatus, 949
  - TPCause, 949
  - userData, 949
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind
  - sms.h, 1492
- unpack\_sms\_SLQSWmsMemoryFullCallBack\_ind\_t, 949
  - messageMode, 950
  - ParamPresenceMask, 950
  - storageType, 950
- unpack\_sms\_SLQSWmsMessageWaitingCallBack\_ind
  - sms.h, 1492
- unpack\_sms\_SLQSWmsMessageWaitingCallBack\_ind\_t, 950
  - msgWaitInfo, 950
  - numInstances, 950
  - ParamPresenceMask, 950
- unpack\_sms\_SaveSMS
  - sms.h, 1482
- unpack\_sms\_SaveSMS\_t, 932
  - pMessageIndex, 933
  - ParamPresenceMask, 933
- unpack\_sms\_SendSMS
  - sms.h, 1483
- unpack\_sms\_SendSMS\_t, 933
  - messageFailureCode, 933
  - messageID, 933
  - ParamPresenceMask, 933



- resultcode, 961
- unpack\_swiavms\_SLQSAVMSSessionGetInfo
  - swiavms.h, 1512
- unpack\_swiavms\_SLQSAVMSSessionGetInfo\_t, 961
  - pConfig, 961
  - pNotification, 962
  - pPackageID, 962
  - ParamPresenceMask, 961
  - resultcode, 962
- unpack\_swiavms\_SLQSAVMSSetSettings
  - swiavms.h, 1513
- unpack\_swiavms\_SLQSAVMSSetSettings\_t, 962
  - ParamPresenceMask, 962
  - resultcode, 962
- unpack\_swiavms\_SLQSAVMSSetSettings\_v2
  - swiavms.h, 1514
- unpack\_swiavms\_SLQSAVMSSetSettings\_v2\_t, 963
  - ParamPresenceMask, 963
  - resultcode, 963
- unpack\_swiavms\_SLQSAVMSStartSession
  - swiavms.h, 1514
- unpack\_swiavms\_SLQSAVMSStartSession\_t, 963
  - ParamPresenceMask, 963
  - resultcode, 963
  - sessionResponse, 964
- unpack\_swiavms\_SLQSAVMSStopSession
  - swiavms.h, 1515
- unpack\_swiavms\_SLQSAVMSStopSession\_avc2
  - swiavms.h, 1515
- unpack\_swiavms\_SLQSAVMSStopSession\_avc2\_t
  - swiavms.h, 1504
- unpack\_swiavms\_SLQSAVMSStopSession\_t, 964
  - ParamPresenceMask, 964
  - resultcode, 964
- unpack\_swiavms\_SLQSAvmsSetEventReport
  - swiavms.h, 1513
- unpack\_swiavms\_SLQSAvmsSetEventReport\_t, 962
  - ParamPresenceMask, 962
  - resultcode, 962
- unpack\_swidms\_SLQSSwiDmsGetHWWatchdog
  - swidms.h, 1522
- unpack\_swidms\_SLQSSwiDmsGetHWWatchdog\_t, 964
  - pHWWatchdog, 964
  - ParamPresenceMask, 964
- unpack\_swidms\_SLQSSwiDmsGetMTU
  - swidms.h, 1522
- unpack\_swidms\_SLQSSwiDmsGetMTU\_t, 965
  - pMTUSize3gpp, 965
  - ParamPresenceMask, 965
- unpack\_swidms\_SLQSSwiDmsGetSecureInfo
  - swidms.h, 1522
- unpack\_swidms\_SLQSSwiDmsGetSecureInfo\_t, 965
  - jtagAccessAllowed, 966
  - memoryDumpAllowed, 966
  - ParamPresenceMask, 966
  - secureBootEnabled, 966
  - TlvResult, 966
- unpack\_swidms\_SLQSSwiDmsGetUsbComp
  - swidms.h, 1523
- unpack\_swidms\_SLQSSwiDmsGetUsbComp\_t, 966
  - pInterfaceCfg, 967
  - pSupportedBitmasks, 967
  - ParamPresenceMask, 967
- unpack\_swidms\_SLQSSwiDmsGetUsbNetNum
  - swidms.h, 1523
- unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t, 967
  - ParamPresenceMask, 967
  - usbNetNum, 967
- unpack\_swidms\_SLQSSwiDmsSetHWWatchdog
  - swidms.h, 1524
- unpack\_swidms\_SLQSSwiDmsSetHWWatchdog\_t, 968
  - ParamPresenceMask, 968
  - Tlvresult, 968
- unpack\_swidms\_SLQSSwiDmsSetMTU
  - swidms.h, 1524
- unpack\_swidms\_SLQSSwiDmsSetMTU\_t, 968
  - ParamPresenceMask, 968
  - Tlvresult, 968
- unpack\_swidms\_SLQSSwiDmsSetUsbComp
  - swidms.h, 1524
- unpack\_swidms\_SLQSSwiDmsSetUsbComp\_t, 969
  - ParamPresenceMask, 969
  - Tlvresult, 969
- unpack\_swidms\_SLQSSwiDmsSetUsbNetNum
  - swidms.h, 1525
- unpack\_swidms\_SLQSSwiDmsSetUsbNetNum\_t
  - swidms.h, 1519
- unpack\_swiloc\_SwiLocGetAutoStart
  - swiloc.h, 1526
- unpack\_swiloc\_SwiLocGetAutoStart\_t, 969
  - fix\_rate, 971
  - fix\_rate\_reported, 971
  - fix\_type, 971
  - fix\_type\_reported, 971
  - function, 971
  - function\_reported, 971
  - max\_dist, 971
  - max\_dist\_reported, 971
  - max\_time, 971
  - max\_time\_reported, 971
  - ParamPresenceMask, 971
- unpack\_swiloc\_SwiLocSetAutoStart
  - swiloc.h, 1527
- unpack\_swiloc\_SwiLocSetAutoStart\_t
  - swiloc.h, 1525
- unpack\_swioma\_SLQSOMADMAAlertCallback
  - swioma.h, 1533
- unpack\_swioma\_SLQSOMADMAAlertCallback\_ind
  - swioma.h, 1533
- unpack\_swioma\_SLQSOMADMAAlertCallback\_ind\_t, 971
  - eventType, 972
- unpack\_swioma\_SLQSOMADMAAlertCallback\_t
  - swioma.h, 1528
- unpack\_swioma\_SLQSOMADMCancelSession
  - swioma.h, 1534

- unpack\_swisma\_SLQSOMADMCancelSession\_t  
swisma.h, [1528](#)
- unpack\_swisma\_SLQSOMADMCancelSessionExt  
swismaext.h, [1541](#)
- unpack\_swisma\_SLQSOMADMCancelSessionExt\_t  
swismaext.h, [1537](#)
- unpack\_swisma\_SLQSOMADMGetSessionInfo  
swisma.h, [1534](#)
- unpack\_swisma\_SLQSOMADMGetSessionInfo\_t, [972](#)  
Date, [974](#)  
DateLength, [974](#)  
ParamPresenceMask, [975](#)  
PkgDescLength, [975](#)  
PkgDescription, [975](#)  
PkgName, [975](#)  
PkgNameLength, [975](#)  
RetryCount, [975](#)  
SessionState, [975](#)  
SessionType, [975](#)  
Severity, [975](#)  
Source, [975](#)  
SourceLength, [975](#)  
Status, [975](#)  
Time, [975](#)  
TimeLength, [975](#)  
UpdateCompleteStatus, [975](#)
- unpack\_swisma\_SLQSOMADMGetSessionInfoExt  
swismaext.h, [1541](#)
- unpack\_swisma\_SLQSOMADMGetSessionInfoExt\_t,  
[975](#)  
fumoState, [978](#)  
hfaStatus, [978](#)  
pkgDate, [978](#)  
pkgDesc, [978](#)  
pkgName, [978](#)  
pkgSize, [978](#)  
sessionState, [978](#)  
status, [978](#)
- unpack\_swisma\_SLQSOMADMGetSettings  
swisma.h, [1535](#)
- unpack\_swisma\_SLQSOMADMGetSettings\_t, [979](#)  
Autosdm, [980](#)  
FOTAdownload, [980](#)  
FwAutoCheck, [980](#)  
ParamPresenceMask, [980](#)
- unpack\_swisma\_SLQSOMADMSendSelection  
swisma.h, [1535](#)
- unpack\_swisma\_SLQSOMADMSendSelection\_t  
swisma.h, [1528](#)
- unpack\_swisma\_SLQSOMADMSendSelectionExt  
swismaext.h, [1542](#)
- unpack\_swisma\_SLQSOMADMSendSelectionExt\_t  
swismaext.h, [1538](#)
- unpack\_swisma\_SLQSOMADMSetSettings  
swisma.h, [1536](#)
- unpack\_swisma\_SLQSOMADMSetSettings\_t  
swisma.h, [1528](#)
- unpack\_swisma\_SLQSOMADMSetSettingsExt  
swismaext.h, [1542](#)
- unpack\_swisma\_SLQSOMADMSetSettingsExt\_t  
swismaext.h, [1538](#)
- unpack\_swisma\_SLQSOMADMStartSession  
swisma.h, [1536](#)
- unpack\_swisma\_SLQSOMADMStartSession\_t, [980](#)  
FwAvailability, [981](#)  
ParamPresenceMask, [981](#)
- unpack\_swisma\_SLQSOMADMStartSessionExt  
swismaext.h, [1543](#)
- unpack\_swisma\_SLQSOMADMStartSessionExt\_t  
swismaext.h, [1538](#)
- unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl  
tmd.h, [1547](#)
- unpack\_tmd\_SLQSTmdDeRegNotMitigationLvl\_t, [981](#)  
ParamPresenceMask, [981](#)  
Tlvresult, [981](#)
- unpack\_tmd\_SLQSTmdGetMitigationDevList  
tmd.h, [1547](#)
- unpack\_tmd\_SLQSTmdGetMitigationDevList\_t, [981](#)  
MitigationDevList, [982](#)  
MitigationDevListLen, [982](#)  
ParamPresenceMask, [982](#)  
Tlvresult, [982](#)
- unpack\_tmd\_SLQSTmdGetMitigationLvl  
tmd.h, [1548](#)
- unpack\_tmd\_SLQSTmdGetMitigationLvl\_t, [982](#)  
CurrentmitigationLvl, [983](#)  
ParamPresenceMask, [983](#)  
ReqMitigationLvl, [983](#)  
Tlvresult, [983](#)
- unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind  
tmd.h, [1548](#)
- unpack\_tmd\_SLQSTmdMitigationLvlRptCallback\_ind\_t,  
[983](#)  
deviceId, [983](#)  
deviceIdLen, [983](#)  
lvl, [983](#)  
ParamPresenceMask, [983](#)
- unpack\_tmd\_SLQSTmdRegNotMitigationLvl  
tmd.h, [1548](#)
- unpack\_tmd\_SLQSTmdRegNotMitigationLvl\_t, [983](#)  
ParamPresenceMask, [984](#)  
Tlvresult, [984](#)
- unpack\_uim\_ChangePin  
uim.h, [1563](#)
- unpack\_uim\_ChangePin\_t, [984](#)  
pEncryptedPIN1, [985](#)  
pIndicationToken, [985](#)  
pRemainingRetries, [985](#)  
ParamPresenceMask, [985](#)  
Tlvresult, [985](#)
- unpack\_uim\_GetCardStatus  
uim.h, [1563](#)
- unpack\_uim\_GetCardStatus\_t, [985](#)  
pCardStatus, [985](#)  
pHotSwapStatus, [985](#)  
ParamPresenceMask, [985](#)



- Tlvresult, [985](#)
- unpack\_uim\_GetCardStatusV2
  - uim.h, [1564](#)
- unpack\_uim\_GetCardStatusV2\_t, [986](#)
  - pCardStatus, [986](#)
  - pHotSwapStatus, [986](#)
  - pSimBusyStatus, [986](#)
  - pValidCardStatus, [986](#)
  - ParamPresenceMask, [986](#)
  - Tlvresult, [986](#)
- unpack\_uim\_ReadTransparent
  - uim.h, [1564](#)
- unpack\_uim\_ReadTransparent\_t, [987](#)
  - pCardResult, [987](#)
  - pEncryptedData, [987](#)
  - pIndicationToken, [987](#)
  - pReadResult, [987](#)
  - ParamPresenceMask, [987](#)
  - Tlvresult, [987](#)
- unpack\_uim\_SLQSUIMAuthenticate
  - uim.h, [1565](#)
- unpack\_uim\_SLQSUIMAuthenticate\_t, [989](#)
  - pAuthenticateResult, [990](#)
  - pCardResult, [990](#)
  - pIndicationToken, [990](#)
  - ParamPresenceMask, [990](#)
- unpack\_uim\_SLQSUIMDepersonalization
  - uim.h, [1565](#)
- unpack\_uim\_SLQSUIMDepersonalization\_t, [990](#)
  - pRemainingRetries, [990](#)
  - ParamPresenceMask, [990](#)
- unpack\_uim\_SLQSUIMEventRegister
  - uim.h, [1566](#)
- unpack\_uim\_SLQSUIMEventRegister\_t, [990](#)
  - eventMask, [991](#)
  - ParamPresenceMask, [991](#)
- unpack\_uim\_SLQSUIMGetConfiguration
  - uim.h, [1566](#)
- unpack\_uim\_SLQSUIMGetConfiguration\_t, [991](#)
  - pAutoSelection, [992](#)
  - pHaltSubscription, [992](#)
  - pPersonalizationStatus, [992](#)
  - ParamPresenceMask, [992](#)
- unpack\_uim\_SLQSUIMGetFileAttributes
  - uim.h, [1566](#)
- unpack\_uim\_SLQSUIMGetFileAttributes\_t, [992](#)
  - pCardResult, [993](#)
  - pFileAttributes, [993](#)
  - pIndicationToken, [993](#)
  - ParamPresenceMask, [993](#)
- unpack\_uim\_SLQSUIMGetServiceStatus
  - uim.h, [1567](#)
- unpack\_uim\_SLQSUIMGetServiceStatus\_t, [993](#)
  - pFDNStatus, [993](#)
  - pHiddenKeyStatus, [993](#)
  - pIndex, [993](#)
  - ParamPresenceMask, [993](#)
- unpack\_uim\_SLQSUIMGetSlotsStatus
  - uim.h, [1567](#)
- unpack\_uim\_SLQSUIMGetSlotsStatus\_t, [994](#)
  - pNumberOfPhySlot, [994](#)
  - pUimSlotsStatus, [994](#)
  - ParamPresenceMask, [994](#)
- unpack\_uim\_SLQSUIMGetSlotsStatusV2
  - uim.h, [1568](#)
- unpack\_uim\_SLQSUIMGetSlotsStatusV2\_t, [994](#)
  - ParamPresenceMask, [995](#)
- unpack\_uim\_SLQSUIMPowerDown
  - uim.h, [1568](#)
- unpack\_uim\_SLQSUIMPowerDown\_t
  - uim.h, [1553](#)
- unpack\_uim\_SLQSUIMPowerUp
  - uim.h, [1568](#)
- unpack\_uim\_SLQSUIMPowerUp\_t
  - uim.h, [1553](#)
- unpack\_uim\_SLQSUIMReadRecord
  - uim.h, [1569](#)
- unpack\_uim\_SLQSUIMReadRecord\_t, [995](#)
  - pAdditionalReadResult, [996](#)
  - pCardResult, [996](#)
  - pIndicationToken, [996](#)
  - pReadResult, [996](#)
  - ParamPresenceMask, [996](#)
- unpack\_uim\_SLQSUIMRefreshCallback\_Ind
  - uim.h, [1569](#)
- unpack\_uim\_SLQSUIMRefreshCallback\_Ind\_t, [996](#)
  - ParamPresenceMask, [996](#)
  - refreshEvent, [996](#)
  - TlvPresent, [996](#)
- unpack\_uim\_SLQSUIMRefreshComplete
  - uim.h, [1569](#)
- unpack\_uim\_SLQSUIMRefreshComplete\_t
  - uim.h, [1554](#)
- unpack\_uim\_SLQSUIMRefreshGetLastEvent
  - uim.h, [1570](#)
- unpack\_uim\_SLQSUIMRefreshGetLastEvent\_t, [997](#)
  - pRefreshEvent, [997](#)
  - ParamPresenceMask, [997](#)
- unpack\_uim\_SLQSUIMRefreshOK
  - uim.h, [1570](#)
- unpack\_uim\_SLQSUIMRefreshOK\_t
  - uim.h, [1554](#)
- unpack\_uim\_SLQSUIMRefreshRegister
  - uim.h, [1570](#)
- unpack\_uim\_SLQSUIMRefreshRegister\_t
  - uim.h, [1554](#)
- unpack\_uim\_SLQSUIMReset
  - uim.h, [1571](#)
- unpack\_uim\_SLQSUIMReset\_t
  - uim.h, [1554](#)
- unpack\_uim\_SLQSUIMSetServiceStatus
  - uim.h, [1571](#)
- unpack\_uim\_SLQSUIMSetServiceStatus\_t
  - uim.h, [1554](#)
- unpack\_uim\_SLQSUIMSetStatusChangeCallBack\_ind
  - uim.h, [1571](#)

- unpack\_uim\_SLQSUIMSetStatusChangeCallback\_ind\_t, 997
- unpack\_uim\_SLQSUIMSwitchSlot
  - uim.h, 1572
- unpack\_uim\_SLQSUIMSwitchSlot\_t
  - uim.h, 1554
- unpack\_uim\_SLQSUIMWriteRecord
  - uim.h, 1572
- unpack\_uim\_SLQSUIMWriteRecord\_t, 998
  - pCardResult, 998
  - pIndicationToken, 998
  - ParamPresenceMask, 998
- unpack\_uim\_SLQSUIMWriteTransparent
  - uim.h, 1572
- unpack\_uim\_SLQSUIMWriteTransparent\_t, 998
  - pCardResult, 999
  - pIndicationToken, 999
  - ParamPresenceMask, 999
- unpack\_uim\_SetPinProtection
  - uim.h, 1564
- unpack\_uim\_SetPinProtection\_t, 988
  - pEncryptedPIN1, 988
  - pIndicationToken, 988
  - pRemainingRetries, 988
  - ParamPresenceMask, 988
  - Tlvresult, 988
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind
  - uim.h, 1565
- unpack\_uim\_SetUimSlotStatusChangeCallback\_ind\_t, 988
  - bNumberOfPhySlots, 989
  - ParamPresenceMask, 989
  - slotsstatusChange, 989
- unpack\_uim\_UnblockPin
  - uim.h, 1573
- unpack\_uim\_UnblockPin\_t, 999
  - pEncryptedPIN1, 1000
  - pIndicationToken, 1000
  - pRemainingRetries, 1000
  - ParamPresenceMask, 1000
  - Tlvresult, 1000
- unpack\_uim\_UnblockPinV2
  - uim.h, 1573
- unpack\_uim\_UnblockPinV2\_t, 1000
  - pCardResult, 1001
  - pEncryptedPIN1, 1001
  - pIndicationToken, 1001
  - pRemainingRetries, 1001
  - ParamPresenceMask, 1001
  - Tlvresult, 1001
- unpack\_uim\_VerifyPin
  - uim.h, 1573
- unpack\_uim\_VerifyPin\_t, 1001
  - pEncryptedPIN1, 1001
  - pIndicationToken, 1001
  - pRemainingRetries, 1002
  - ParamPresenceMask, 1001
  - Tlvresult, 1002
- unpack\_valid\_nas\_GetCDMANetworkParameters
  - nas.h, 1385
- unpack\_valid\_nas\_SLQSGetServingSystem
  - nas.h, 1385
- unpack\_valid\_nas\_SLQSGetSignalStrength
  - nas.h, 1386
- unpack\_valid\_nas\_SLQSNasGetSigInfo
  - nas.h, 1387
- unpack\_voice\_AnswerUSSD
  - voice.h, 1592
- unpack\_voice\_AnswerUSSD\_t
  - voice.h, 1580
- unpack\_voice\_CancelUSSD
  - voice.h, 1593
- unpack\_voice\_CancelUSSD\_t
  - voice.h, 1580
- unpack\_voice\_DTMFEventCallback\_ind
  - voice.h, 1593
- unpack\_voice\_DTMFEventCallback\_ind\_t, 1004
  - DTMFInformation, 1004
  - pOffLength, 1005
  - pOnLength, 1005
  - ParamPresenceMask, 1004
- unpack\_voice\_OTASPStatusCallback\_ind
  - voice.h, 1594
- unpack\_voice\_OTASPStatusCallback\_ind\_t, 1005
  - callID, 1006
  - OTASPStatus, 1006
  - ParamPresenceMask, 1006
- unpack\_voice\_OriginateUSSD
  - voice.h, 1593
- unpack\_voice\_OriginateUSSD\_t
  - voice.h, 1580
- unpack\_voice\_SLQSOriginateUSSD
  - voice.h, 1594
- unpack\_voice\_SLQSOriginateUSSD\_t, 1006
  - pAlphaIDInfo, 1009
  - pCCSuppsType, 1009
  - pCallId, 1009
  - pCcResultType, 1009
  - pUSSDInfo, 1009
  - ParamPresenceMask, 1009
  - pfailureCause, 1009
- unpack\_voice\_SLQSVoiceALSSelectLine
  - voice.h, 1594
- unpack\_voice\_SLQSVoiceALSSelectLine\_t
  - voice.h, 1580
- unpack\_voice\_SLQSVoiceALSSetLineSwitching
  - voice.h, 1595
- unpack\_voice\_SLQSVoiceALSSetLineSwitching\_t
  - voice.h, 1580
- unpack\_voice\_SLQSVoiceAnswerCall
  - voice.h, 1595
- unpack\_voice\_SLQSVoiceAnswerCall\_t, 1009
  - pCallId, 1010
  - ParamPresenceMask, 1010
- unpack\_voice\_SLQSVoiceBindSubscription
  - voice.h, 1596



- unpack\_voice\_SLQSVoiceBindSubscription\_t  
voice.h, [1580](#)
- unpack\_voice\_SLQSVoiceBurstDTMF  
voice.h, [1596](#)
- unpack\_voice\_SLQSVoiceBurstDTMF\_t, [1010](#)  
pCallID, [1010](#)  
ParamPresenceMask, [1010](#)
- unpack\_voice\_SLQSVoiceDialCall  
voice.h, [1596](#)
- unpack\_voice\_SLQSVoiceDialCall\_t, [1010](#)  
pAlphaIDInfo, [1011](#)  
pCCResultType, [1011](#)  
pCCSUPSType, [1011](#)  
pCallID, [1011](#)  
ParamPresenceMask, [1011](#)
- unpack\_voice\_SLQSVoiceEndCall  
voice.h, [1597](#)
- unpack\_voice\_SLQSVoiceEndCall\_t, [1011](#)  
pCallID, [1012](#)  
ParamPresenceMask, [1012](#)
- unpack\_voice\_SLQSVoiceGetAllCallInfo  
voice.h, [1597](#)
- unpack\_voice\_SLQSVoiceGetAllCallInfo\_t, [1012](#)  
pArrAlertingPattern, [1015](#)  
pArrAlertingType, [1015](#)  
pArrAlphaID, [1015](#)  
pArrCallEndReason, [1015](#)  
pArrCallInfo, [1015](#)  
pArrCalledPartyNum, [1015](#)  
pArrConnectPartyNum, [1015](#)  
pArrDiagInfo, [1015](#)  
pArrRedirPartyNum, [1015](#)  
pArrRemotePartyName, [1015](#)  
pArrRemotePartyNum, [1015](#)  
pArrSvcOption, [1015](#)  
pArrUUSInfo, [1015](#)  
pVoicePrivacy, [1015](#)  
ParamPresenceMask, [1015](#)
- unpack\_voice\_SLQSVoiceGetCLIP  
voice.h, [1599](#)
- unpack\_voice\_SLQSVoiceGetCLIP\_t, [1023](#)  
pAlphaIDInfo, [1024](#)  
pCCResType, [1024](#)  
pCallID, [1024](#)  
pFailCause, [1025](#)  
ParamPresenceMask, [1024](#)
- unpack\_voice\_SLQSVoiceGetCLIR  
voice.h, [1599](#)
- unpack\_voice\_SLQSVoiceGetCLIR\_t, [1025](#)  
pAlphaIDInfo, [1026](#)  
pCCResType, [1026](#)  
pCallID, [1026](#)  
pFailCause, [1026](#)  
ParamPresenceMask, [1026](#)
- unpack\_voice\_SLQSVoiceGetCNAP  
voice.h, [1599](#)
- unpack\_voice\_SLQSVoiceGetCNAP\_t, [1026](#)  
pAlphaIDInfo, [1027](#)  
pCCResType, [1027](#)  
pCallID, [1027](#)  
pFailCause, [1027](#)  
ParamPresenceMask, [1027](#)
- unpack\_voice\_SLQSVoiceGetCOLP  
voice.h, [1600](#)
- unpack\_voice\_SLQSVoiceGetCOLP\_t, [1028](#)  
pAlphaIDInfo, [1029](#)  
pCCResType, [1029](#)  
pCallID, [1029](#)  
pFailCause, [1029](#)  
ParamPresenceMask, [1029](#)
- unpack\_voice\_SLQSVoiceGetCOLR  
voice.h, [1600](#)
- unpack\_voice\_SLQSVoiceGetCOLR\_t, [1029](#)  
pAlphaIDInfo, [1030](#)  
pCCResType, [1030](#)  
pCallID, [1030](#)  
pFailCause, [1030](#)  
ParamPresenceMask, [1030](#)
- unpack\_voice\_SLQSVoiceGetCallBarring  
voice.h, [1597](#)
- unpack\_voice\_SLQSVoiceGetCallBarring\_t, [1016](#)  
pAlphaIDInfo, [1017](#)  
pCCResType, [1017](#)  
pCallID, [1017](#)  
pFailCause, [1017](#)  
pSvcClass, [1017](#)  
ParamPresenceMask, [1017](#)
- unpack\_voice\_SLQSVoiceGetCallForwardingStatus  
voice.h, [1598](#)
- unpack\_voice\_SLQSVoiceGetCallForwardingStatus\_t,  
[1017](#)  
pAlphaIDInfo, [1018](#)  
pCCResType, [1018](#)  
pCallID, [1018](#)  
pFailCause, [1018](#)  
ParamPresenceMask, [1018](#)
- unpack\_voice\_SLQSVoiceGetCallInfo  
voice.h, [1598](#)
- unpack\_voice\_SLQSVoiceGetCallInfo\_t, [1019](#)  
pAlertType, [1021](#)  
pAlertingPattern, [1021](#)  
pAlphaIDInfo, [1021](#)  
pCallInfo, [1021](#)  
pConnectNumInfo, [1021](#)  
pDiagInfo, [1021](#)  
pOTASPStatus, [1021](#)  
pRemotePartyName, [1021](#)  
pRemotePartyNum, [1021](#)  
pSrvOpt, [1021](#)  
pUUSInfo, [1022](#)  
pVoicePrivacy, [1022](#)  
ParamPresenceMask, [1021](#)
- unpack\_voice\_SLQSVoiceGetCallWaiting  
voice.h, [1598](#)
- unpack\_voice\_SLQSVoiceGetCallWaiting\_t, [1022](#)  
pAlphaIDInfo, [1023](#)

pCCResType, 1023  
 pCallID, 1023  
 pFailCause, 1023  
 pSvcClass, 1023  
 ParamPresenceMask, 1023  
 unpack\_voice\_SLQSVoiceGetConfig  
   voice.h, 1600  
 unpack\_voice\_SLQSVoiceGetConfig\_t, 1030  
   pAirTimerCnt, 1032  
   pAutoAnswerStat, 1032  
   pCurAMRConfig, 1032  
   pCurPrefVoiceSO, 1032  
   pCurVoiceDomainPref, 1032  
   pCurVoicePrivacyPref, 1032  
   pCurrTTYMode, 1032  
   pRoamTimerCnt, 1032  
   ParamPresenceMask, 1032  
 unpack\_voice\_SLQSVoiceIndicationRegister  
   voice.h, 1601  
 unpack\_voice\_SLQSVoiceIndicationRegister\_t  
   voice.h, 1580  
 unpack\_voice\_SLQSVoiceManageCalls  
   voice.h, 1601  
 unpack\_voice\_SLQSVoiceManageCalls\_t, 1032  
   pFailCause, 1033  
   ParamPresenceMask, 1033  
 unpack\_voice\_SLQSVoiceOrigUSSDNoWait  
   voice.h, 1601  
 unpack\_voice\_SLQSVoiceOrigUSSDNoWait\_t  
   voice.h, 1580  
 unpack\_voice\_SLQSVoiceSUPSCallback\_ind  
   voice.h, 1604  
 unpack\_voice\_SLQSVoiceSUPSCallback\_ind\_t, 1039  
   pDataSrc, 1042  
   pFailCause, 1042  
   pReason, 1042  
   pSvcClass, 1042  
   ParamPresenceMask, 1041  
 unpack\_voice\_SLQSVoiceSendFlash  
   voice.h, 1602  
 unpack\_voice\_SLQSVoiceSendFlash\_t, 1033  
   pCallID, 1033  
   ParamPresenceMask, 1033  
 unpack\_voice\_SLQSVoiceSetCallBarringPassword  
   voice.h, 1602  
 unpack\_voice\_SLQSVoiceSetCallBarringPassword\_t,  
   1034  
   pAlphaIDInfo, 1035  
   pCCResType, 1035  
   pCallID, 1035  
   pFailCause, 1035  
   ParamPresenceMask, 1035  
 unpack\_voice\_SLQSVoiceSetConfig  
   voice.h, 1602  
 unpack\_voice\_SLQSVoiceSetConfig\_t, 1035  
   pAirTimerStatus, 1036  
   pAutoAnsStatus, 1036  
   pPrefVoiceSOStatus, 1036  
   pRoamTimerStatus, 1036  
   pTTYConfigStatus, 1036  
   pVoiceDomainPrefStatus, 1036  
   ParamPresenceMask, 1036  
 unpack\_voice\_SLQSVoiceSetPreferredPrivacy  
   voice.h, 1603  
 unpack\_voice\_SLQSVoiceSetPreferredPrivacy\_t  
   voice.h, 1580  
 unpack\_voice\_SLQSVoiceSetSUPSService  
   voice.h, 1603  
 unpack\_voice\_SLQSVoiceSetSUPSService\_t, 1037  
   pCallID, 1038  
   pFailCause, 1038  
   ParamPresenceMask, 1038  
 unpack\_voice\_SLQSVoiceStartContDTMF  
   voice.h, 1603  
 unpack\_voice\_SLQSVoiceStartContDTMF\_t, 1038  
   pCallID, 1038  
   ParamPresenceMask, 1038  
 unpack\_voice\_SLQSVoiceStopContDTMF  
   voice.h, 1604  
 unpack\_voice\_SLQSVoiceStopContDTMF\_t, 1038  
   callID, 1039  
   ParamPresenceMask, 1039  
 unpack\_voice\_SUPSNotificationCallback\_ind  
   voice.h, 1604  
 unpack\_voice\_SUPSNotificationCallback\_ind\_t, 1042  
   callID, 1044  
   notifType, 1044  
   pCUGIndex, 1044  
   pECTNum, 1044  
   ParamPresenceMask, 1044  
 unpack\_voice\_USSDNotificationCallback\_ind  
   voice.h, 1605  
 unpack\_voice\_USSDNotificationCallback\_ind\_t, 1044  
   notification\_Type, 1044  
   ParamPresenceMask, 1044  
 unpack\_voice\_VoiceInfoRecCallback\_ind  
   voice.h, 1605  
 unpack\_voice\_VoiceInfoRecCallback\_ind\_t, 1044  
   callID, 1047  
   pCLIRCause, 1047  
   pCallWaitInd, 1047  
   pCalledPartyInfo, 1047  
   pCallerIDInfo, 1047  
   pCallerNameInfo, 1047  
   pCallingPartyInfo, 1047  
   pConnectNumInfo, 1047  
   pDispInfo, 1047  
   pExtDispInfo, 1047  
   pExtDispRecInfo, 1047  
   pLineCtrlInfo, 1047  
   pNSSAudioCtrl, 1047  
   pNSSRelease, 1047  
   pRedirNumInfo, 1047  
   pSignalInfo, 1047  
   ParamPresenceMask, 1047  
 unpack\_voice\_allCallStatusCallback\_ind

- voice.h, [1592](#)
- unpack\_voice\_allCallStatusCallback\_ind\_t, [1002](#)
  - arrCallInformation, [1003](#)
  - pArrAlertingPattern, [1003](#)
  - pArrAlertingType, [1003](#)
  - pArrAlphaID, [1003](#)
  - pArrCallEndReason, [1003](#)
  - pArrCalledPartyNum, [1003](#)
  - pArrConnectPartyNum, [1003](#)
  - pArrDiagInfo, [1003](#)
  - pArrRedirPartyNum, [1003](#)
  - pArrRemotePartyName, [1003](#)
  - pArrRemotePartyNum, [1003](#)
  - pArrSvcOption, [1003](#)
  - ParamPresenceMask, [1003](#)
- unpack\_voice\_voicePrivacyChangeCallback\_ind
  - voice.h, [1605](#)
- unpack\_voice\_voicePrivacyChangeCallback\_ind\_t,
  - [1047](#)
  - callID, [1048](#)
  - ParamPresenceMask, [1048](#)
  - voicePrivacy, [1048](#)
- unpack\_wds\_DHCPv4ClientLease\_ind
  - wds.h, [1637](#)
- unpack\_wds\_DHCPv4ClientLease\_ind\_t, [1048](#)
  - DHCPv4LeaseOptTlv, [1049](#)
  - DHCPv4LeaseStateTlv, [1049](#)
  - IPv4AddrTlv, [1049](#)
  - ParamPresenceMask, [1049](#)
  - ProfileIdTlv, [1049](#)
- unpack\_wds\_DHCPv4ClientLeaseChange
  - wds.h, [1638](#)
- unpack\_wds\_DHCPv4ClientLeaseChange\_t
  - wds.h, [1615](#)
- unpack\_wds\_GetAutoconnect
  - wds.h, [1638](#)
- unpack\_wds\_GetAutoconnect\_t, [1049](#)
  - ParamPresenceMask, [1049](#)
  - psetting, [1049](#)
- unpack\_wds\_GetByteTotals
  - wds.h, [1639](#)
- unpack\_wds\_GetByteTotals\_t, [1049](#)
  - pRXTotalBytes, [1050](#)
  - pTXTotalBytes, [1050](#)
  - ParamPresenceMask, [1050](#)
- unpack\_wds\_GetConnectionRate
  - wds.h, [1639](#)
- unpack\_wds\_GetConnectionRate\_t, [1050](#)
  - currentChannelRXRate, [1051](#)
  - currentChannelTXRate, [1051](#)
  - maxChannelRXRate, [1051](#)
  - maxChannelTXRate, [1051](#)
  - ParamPresenceMask, [1051](#)
- unpack\_wds\_GetDataBearerTechnology
  - wds.h, [1639](#)
- unpack\_wds\_GetDataBearerTechnology\_t, [1051](#)
  - pDataBearer, [1052](#)
  - ParamPresenceMask, [1052](#)
- unpack\_wds\_GetDefaultProfile
  - wds.h, [1640](#)
- unpack\_wds\_GetDefaultProfile\_t, [1052](#)
  - apnname, [1054](#)
  - apnsize, [1054](#)
  - auth, [1054](#)
  - ipaddr, [1054](#)
  - ipaddrv6, [1055](#)
  - name, [1055](#)
  - namesize, [1055](#)
  - ParamPresenceMask, [1055](#)
  - pdptype, [1055](#)
  - pridns, [1055](#)
  - pridnsv6, [1055](#)
  - secdns, [1055](#)
  - secdnsv6, [1055](#)
  - username, [1055](#)
  - usersize, [1055](#)
- unpack\_wds\_GetDefaultProfileNum
  - wds.h, [1640](#)
- unpack\_wds\_GetDefaultProfileNum\_t, [1055](#)
  - index, [1055](#)
  - ParamPresenceMask, [1055](#)
- unpack\_wds\_GetDefaultProfileV2
  - wds.h, [1640](#)
- unpack\_wds\_GetDefaultProfileV2\_t, [1055](#)
  - apnname, [1058](#)
  - apnsize, [1058](#)
  - auth, [1058](#)
  - ipaddr, [1058](#)
  - ipaddrv6, [1058](#)
  - name, [1058](#)
  - namesize, [1058](#)
  - ParamPresenceMask, [1058](#)
  - pdptype, [1058](#)
  - pridns, [1058](#)
  - pridnsv6, [1058](#)
  - pwd, [1058](#)
  - pwdsize, [1058](#)
  - secdns, [1058](#)
  - secdnsv6, [1058](#)
  - username, [1058](#)
  - usersize, [1058](#)
- unpack\_wds\_GetDormancyState
  - wds.h, [1641](#)
- unpack\_wds\_GetDormancyState\_t, [1058](#)
  - dormancyState, [1059](#)
  - ParamPresenceMask, [1059](#)
- unpack\_wds\_GetLastMobileIPError
  - wds.h, [1641](#)
- unpack\_wds\_GetLastMobileIPError\_t, [1059](#)
  - error, [1059](#)
  - ParamPresenceMask, [1059](#)
- unpack\_wds\_GetMobileIP
  - wds.h, [1641](#)
- unpack\_wds\_GetMobileIP\_t, [1060](#)
  - mipMode, [1060](#)
  - ParamPresenceMask, [1060](#)

- unpack\_wds\_GetMobileIPProfile
  - wds.h, 1642
- unpack\_wds\_GetMobileIPProfile\_t, 1060
  - AAASPI, 1062
  - AAASState, 1062
  - address, 1062
  - enabled, 1062
  - HASPI, 1062
  - HASState, 1062
  - NAI, 1062
  - naiSize, 1062
  - ParamPresenceMask, 1062
  - primaryHA, 1062
  - revTunneling, 1062
  - secondaryHA, 1062
- unpack\_wds\_GetPacketStatistics
  - wds.h, 1642
- unpack\_wds\_GetPacketStatistics\_t, 1062
  - pRXDroppedCount, 1064
  - pRXOKBytesLastCall, 1064
  - pRXOkBytesCount, 1064
  - pRXPacketErrors, 1064
  - pRXPacketOverflows, 1064
  - pRXPacketSuccesses, 1064
  - pTXDroppedCount, 1064
  - pTXOKBytesLastCall, 1064
  - pTXOkBytesCount, 1064
  - pTXPacketErrors, 1064
  - pTXPacketOverflows, 1064
  - pTXPacketSuccesses, 1064
  - ParamPresenceMask, 1064
- unpack\_wds\_GetPacketStatus
  - wds.h, 1642
- unpack\_wds\_GetPacketStatus\_t, 1064
  - ParamPresenceMask, 1066
  - rXDroppedCount, 1066
  - rXOKBytesLastCall, 1066
  - rXOkBytesCount, 1066
  - rXPacketErrors, 1066
  - rXPacketOverflows, 1066
  - rXPacketSuccesses, 1066
  - tXDroppedCount, 1066
  - tXOKBytesLastCall, 1066
  - tXOkBytesCount, 1066
  - tXPacketErrors, 1066
  - tXPacketOverflows, 1066
  - tXPacketSuccesses, 1066
- unpack\_wds\_GetSessionDuration
  - wds.h, 1643
- unpack\_wds\_GetSessionDuration\_t, 1066
  - callDuration, 1067
  - ParamPresenceMask, 1067
- unpack\_wds\_GetSessionDurationV2
  - wds.h, 1643
- unpack\_wds\_GetSessionDurationV2\_t, 1067
  - callDuration, 1068
  - pCallActiveDuration, 1068
  - pLastCallActiveDuration, 1068
  - pLastCallDuration, 1068
  - ParamPresenceMask, 1068
- unpack\_wds\_GetSessionState
  - wds.h, 1643
- unpack\_wds\_GetSessionState\_t, 1068
  - connectionStatus, 1068
  - ParamPresenceMask, 1068
- unpack\_wds\_RMSetTransferStatistics
  - wds.h, 1644
- unpack\_wds\_RMSetTransferStatistics\_t, 1068
  - ParamPresenceMask, 1069
- unpack\_wds\_RMTransferStatistics\_ind
  - wds.h, 1644
- unpack\_wds\_RMTransferStatistics\_ind\_t
  - wds.h, 1615
- unpack\_wds\_SLQSCreateProfile
  - wds.h, 1647
- unpack\_wds\_SLQSCreateProfile\_t, 1069
  - pCreateProfileOut, 1070
  - pProfileID, 1070
  - ParamPresenceMask, 1070
  - Tlvresult, 1070
- unpack\_wds\_SLQSDUNCallInfoCallBack\_ind
  - wds.h, 1647
- unpack\_wds\_SLQSDUNCallInfoCallBack\_ind\_t, 1070
- unpack\_wds\_SLQSDDeleteProfile
  - wds.h, 1647
- unpack\_wds\_SLQSDDeleteProfile\_t, 1070
  - extendedErrorCode, 1070
  - ParamPresenceMask, 1070
- unpack\_wds\_SLQSGet3GPPConfigItem
  - wds.h, 1648
- unpack\_wds\_SLQSGet3GPPConfigItem\_t, 1072
  - \_3gppRelease, 1073
  - defaultPDNEnabled, 1073
  - LTEAttachProfile, 1073
  - ParamPresenceMask, 1074
  - profileList, 1074
- unpack\_wds\_SLQSGetCurrDataSystemStat
  - wds.h, 1648
- unpack\_wds\_SLQSGetCurrDataSystemStat\_t, 1074
  - currNetworkInfo, 1074
  - networkInfoLen, 1074
  - ParamPresenceMask, 1074
  - prefNetwork, 1075
- unpack\_wds\_SLQSGetCurrentChannelRate
  - wds.h, 1648
- unpack\_wds\_SLQSGetCurrentChannelRate\_t, 1075
  - max\_channel\_rx\_rate, 1076
  - max\_channel\_tx\_rate, 1076
  - ParamPresenceMask, 1076
- unpack\_wds\_SLQSGetDUNCallInfo
  - wds.h, 1649
- unpack\_wds\_SLQSGetDUNCallInfo\_t, 1077
  - callEndReason, 1079
  - channelRate, 1079
  - connectionStatus, 1079
  - dataBearerTech, 1079

- dormancyStatus, [1079](#)
- lastCallDataBearerTech, [1079](#)
- mdmCallDurationActive, [1080](#)
- ParamPresenceMask, [1080](#)
- rxOKBytesCount, [1080](#)
- txOKBytesCount, [1080](#)
- unpack\_wds\_SLQSSGetDataBearerTechnology
  - wds.h, [1649](#)
- unpack\_wds\_SLQSSGetDataBearerTechnology\_t, [1076](#)
  - curDataBearerTechnology, [1076](#)
  - dataBearerMask, [1076](#)
  - lastCallDataBearerTechnology, [1077](#)
  - ParamPresenceMask, [1077](#)
- unpack\_wds\_SLQSSGetProfileSettings
  - wds.h, [1649](#)
- unpack\_wds\_SLQSSGetProfileSettings\_t, [1080](#)
  - pProfileSettings, [1080](#)
  - ParamPresenceMask, [1080](#)
  - ProfileType, [1080](#)
  - Tlvresult, [1080](#)
- unpack\_wds\_SLQSSGetProfileSettingsV2
  - wds.h, [1650](#)
- unpack\_wds\_SLQSSGetProfileSettingsV2\_t, [1080](#)
  - pProfileSettings, [1081](#)
  - ParamPresenceMask, [1081](#)
  - ProfileType, [1081](#)
  - Tlvresult, [1081](#)
- unpack\_wds\_SLQSSGetRuntimeSettings
  - wds.h, [1650](#)
- unpack\_wds\_SLQSSGetRuntimeSettings\_t, [1081](#)
  - APNName, [1083](#)
  - Authentication, [1083](#)
  - DomainList, [1083](#)
  - GPRSGrantedQoS, [1083](#)
  - GWAddressV4, [1083](#)
  - IMCNflag, [1084](#)
  - IPFamilyPreference, [1084](#)
  - IPv6AddrInfo, [1084](#)
  - IPv6GWAddrInfo, [1084](#)
  - IPv4, [1084](#)
  - Mtu, [1084](#)
  - PDPTType, [1084](#)
  - ParamPresenceMask, [1084](#)
  - PrimaryDNSV4, [1084](#)
  - PrimaryDNSV6, [1084](#)
  - ProfileID, [1084](#)
  - ProfileName, [1084](#)
  - SecondaryDNSV4, [1084](#)
  - SecondaryDNSV6, [1084](#)
  - ServerAddrList, [1084](#)
  - SubnetMaskV4, [1084](#)
  - Technology, [1084](#)
  - UMTSGrantedQoS, [1084](#)
  - Username, [1084](#)
- unpack\_wds\_SLQSSModifyProfile
  - wds.h, [1650](#)
- unpack\_wds\_SLQSSModifyProfile\_t, [1084](#)
  - pExtErrorCode, [1085](#)
  - ParamPresenceMask, [1085](#)
- unpack\_wds\_SLQSSResetPacketStatics
  - wds.h, [1651](#)
- unpack\_wds\_SLQSSResetPacketStatics\_t
  - wds.h, [1615](#)
- unpack\_wds\_SLQSSGetDHCPv4ClientConfig
  - wds.h, [1653](#)
- unpack\_wds\_SLQSSGetDHCPv4ClientConfig\_t, [1093](#)
  - pHwConfig, [1094](#)
  - ParamPresenceMask, [1094](#)
- unpack\_wds\_SLQSSGetLoopback
  - wds.h, [1653](#)
- unpack\_wds\_SLQSSGetLoopback\_t, [1094](#)
  - ByteLoopbackMode, [1095](#)
  - ByteLoopbackMultiplier, [1095](#)
  - ParamPresenceMask, [1095](#)
- unpack\_wds\_SLQSSSetDHCPv4ClientConfig
  - wds.h, [1653](#)
- unpack\_wds\_SLQSSSetDHCPv4ClientConfig\_t
  - wds.h, [1615](#)
- unpack\_wds\_SLQSSSetLoopback
  - wds.h, [1654](#)
- unpack\_wds\_SLQSSSetLoopback\_t
  - wds.h, [1615](#)
- unpack\_wds\_SLQSSSet3GPPConfigItem
  - wds.h, [1651](#)
- unpack\_wds\_SLQSSSet3GPPConfigItem\_t
  - wds.h, [1615](#)
- unpack\_wds\_SLQSSSetIPFamilyPreference
  - wds.h, [1651](#)
- unpack\_wds\_SLQSSSetIPFamilyPreference\_t, [1085](#)
  - ParamPresenceMask, [1085](#)
  - Tlvresult, [1085](#)
- unpack\_wds\_SLQSSSetPacketSrvStatusCallback
  - wds.h, [1652](#)
- unpack\_wds\_SLQSSSetPacketSrvStatusCallback\_t, [1085](#)
  - bearerID, [1087](#)
  - conn\_status, [1087](#)
  - ipFamily, [1087](#)
  - ParamPresenceMask, [1087](#)
  - reconfigReqd, [1087](#)
  - sessionEndReason, [1087](#)
  - techName, [1087](#)
  - verboseSessnEndReason, [1087](#)
  - verboseSessnEndReasonType, [1087](#)
- unpack\_wds\_SLQSSSetWdsEventCallback
  - wds.h, [1652](#)
- unpack\_wds\_SLQSSSetWdsEventCallback\_ind
  - wds.h, [1652](#)
- unpack\_wds\_SLQSSSetWdsEventCallback\_ind\_t, [1087](#)
  - currDBTechAvail, [1092](#)
  - currNWInfo, [1092](#)
  - dBTechAvail, [1092](#)
  - dBTechnology, [1093](#)
  - dBtechExtAvail, [1092](#)
  - dBtechnologyExt, [1093](#)
  - dataSysStatAvail, [1092](#)

- dormancyStatAvail, 1093
- dormancyStatus, 1093
- mipStatus, 1093
- mipstatAvail, 1093
- netInfoLen, 1093
- ParamPresenceMask, 1093
- prefNetwork, 1093
- ratMask, 1093
- rx\_bytes, 1093
- rx\_pkts, 1093
- soMask, 1093
- tx\_bytes, 1093
- tx\_pkts, 1093
- xferStatAvail, 1093
- unpack\_wds\_SLQSSetWdsEventCallback\_t
  - wds.h, 1615
- unpack\_wds\_SLQSStartDataSession
  - wds.h, 1654
- unpack\_wds\_SLQSStartDataSession\_t, 1095
  - pFailureReason, 1095
  - pVerboseFailReasonType, 1096
  - pVerboseFailureReason, 1096
  - ParamPresenceMask, 1095
  - psid, 1095
- unpack\_wds\_SLQSStopDataSession
  - wds.h, 1654
- unpack\_wds\_SLQSStopDataSession\_t
  - wds.h, 1615
- unpack\_wds\_SLQSSwiProfileChangeCallback
  - wds.h, 1655
- unpack\_wds\_SLQSSwiProfileChangeCallback\_Ind
  - wds.h, 1655
- unpack\_wds\_SLQSSwiProfileChangeCallback\_Ind\_t, 1096
  - ParamPresenceMask, 1096
  - ProfileTlv, 1096
  - srcTlv, 1096
- unpack\_wds\_SLQSSwiProfileChangeCallback\_t
  - wds.h, 1615
- unpack\_wds\_SLQSWdsGoActive
  - wds.h, 1655
- unpack\_wds\_SLQSWdsGoActive\_t
  - wds.h, 1615
- unpack\_wds\_SLQSWdsGoDormant
  - wds.h, 1656
- unpack\_wds\_SLQSWdsGoDormant\_t
  - wds.h, 1615
- unpack\_wds\_SLQSWdsSetEventReport
  - wds.h, 1656
- unpack\_wds\_SLQSWdsSetEventReport\_t
  - wds.h, 1615
- unpack\_wds\_SLQSWdsSwdPDRuntimeSettings
  - wds.h, 1656
- unpack\_wds\_SLQSWdsSwdPDRuntimeSettings\_t, 1096
  - apnName, 1098
  - bearerId, 1098
  - contextId, 1098
  - ipv4Address, 1098
  - ipv4GWAddress, 1098
  - ipv6Address, 1098
  - ipv6GWAddress, 1098
  - ParamPresenceMask, 1098
- unpack\_wds\_SetAutoconnect
  - wds.h, 1644
- unpack\_wds\_SetAutoconnect\_t
  - wds.h, 1615
- unpack\_wds\_SetDefaultProfile
  - wds.h, 1645
- unpack\_wds\_SetDefaultProfile\_t
  - wds.h, 1615
- unpack\_wds\_SetDefaultProfileNum
  - wds.h, 1645
- unpack\_wds\_SetDefaultProfileNum\_t
  - wds.h, 1615
- unpack\_wds\_SetMobileIP
  - wds.h, 1645
- unpack\_wds\_SetMobileIP\_t
  - wds.h, 1615
- unpack\_wds\_SetMobileIPParameters
  - wds.h, 1646
- unpack\_wds\_SetMobileIPParameters\_t
  - wds.h, 1615
- unpack\_wds\_SetMobileIPProfile
  - wds.h, 1646
- unpack\_wds\_SetMobileIPProfile\_t, 1069
  - ParamPresenceMask, 1069
- unpack\_wds\_SetMuxID
  - wds.h, 1646
- unpack\_wds\_SetMuxID\_t
  - wds.h, 1615
- UnpackQmiProfileInfo
  - wds.h, 1615
- UnpackQmiProfileInfoV2
  - wds.h, 1616
- UnpackSwdAvmsEventReportBinaryUpdateSessionInfo, 1100
  - bBinaryType, 1101
  - bSentry, 1101
  - bState, 1101
  - bUserInputRequest, 1101
  - szDescription, 1101
  - szName, 1101
  - szVersion, 1102
  - TlvPresent, 1102
  - ulPkgDownloadComplete, 1102
  - ulPkgDownloadSize, 1102
  - wDescriptionLength, 1102
  - wNameLength, 1102
  - wUpdateCompeteStatus, 1102
  - wUserInputTimeout, 1102
  - wVersionLength, 1102
- UnpackSwdAvmsEventReportConfig, 1102
  - bState, 1103
  - bUserInputRequest, 1103
  - szAlertMsg, 1103



- TlvPresent, 1103
- wAlertMsgLength, 1103
- wUserInputTimeout, 1103
- UnpackSwiAvmsEventReportConnectionRequest, 1103
  - bUserInputRequest, 1103
  - TlvPresent, 1103
  - wUserInputTimeout, 1103
- UnpackSwiAvmsEventReportDataSessionStatus, 1103
  - bType, 1104
  - TlvPresent, 1104
  - wErrorCode, 1104
- UnpackSwiAvmsEventReportHTTPStatus, 1104
  - TlvPresent, 1105
  - wHTTPStatus, 1105
- UnpackSwiAvmsEventReportNotification, 1106
  - bNotification, 1106
  - TlvPresent, 1107
  - wSessionStatus, 1107
- UnpackSwiAvmsEventReportPackageID, 1107
  - bPackageID, 1107
  - TlvPresent, 1107
- UnpackSwiAvmsEventReportRegStatus, 1107
  - bRegStatus, 1107
  - TlvPresent, 1107
- UnpackSwiAvmsEventReportSessionType, 1108
  - bType, 1108
  - TlvPresent, 1108
- UnpackSwiAvmsEventReportWAMSPParamChange, 1108
  - TlvPresent, 1109
  - wWamsChangeMask, 1109
- unpackWdsProfileParam, 1109
  - SlqsProfile3GPP, 1109
  - SlqsProfile3GPP2, 1109
- unpackWdsProfileParamV2, 1109
  - SlqsProfile3GPP, 1110
  - SlqsProfile3GPP2, 1110
- upLink
  - voice\_NSSAudioCtrl, 1146
- UpdateCompleteStatus
  - unpack\_swima\_SLQSOMADMGetSessionInfo\_t, 975
- updateCompleteStatus
  - unpack\_omaDmFotaTlv\_t, 891
- upinRetries
  - slotInf, 594
  - uim\_slotInfo, 649
- upinState
  - slotInf, 594
  - uim\_slotInfo, 649
- upukRetries
  - slotInf, 594
  - uim\_slotInfo, 649
- urlAddr
  - loc\_URLAddrInfo, 169
- usageMask
  - loc\_sensorDataUsage, 166
- UsageSetting
  - nas\_UsageSettingTlv, 344
- UsbMTUSize
  - swidms\_usbMTUSizeTlv, 618
- usbNetNum
  - unpack\_swidms\_SLQSSwiDmsGetUsbNetNum\_t, 967
- userData
  - unpack\_sms\_SLQSWmsAsyncRawSendCallBack\_ind\_t, 949
- userInputReq
  - unpack\_omaDmConfigTlv\_t, 889
  - unpack\_omaDmFotaTlv\_t, 891
- userInputTimeout
  - unpack\_omaDmConfigTlv\_t, 889
  - unpack\_omaDmFotaTlv\_t, 891
- Username
  - unpack\_wds\_SLQSGetRuntimeSettings\_t, 1084
- username
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- usersize
  - unpack\_wds\_GetDefaultProfile\_t, 1055
  - unpack\_wds\_GetDefaultProfileV2\_t, 1058
- ussDCS
  - pack\_voice\_SLQSOriginateUSSD\_t, 537
  - voice\_USSInfo, 1156
- ussData
  - pack\_voice\_SLQSOriginateUSSD\_t, 537
  - voice\_USSInfo, 1156
- ussLen
  - pack\_voice\_SLQSOriginateUSSD\_t, 537
  - voice\_USSInfo, 1156
- UtRat
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- UtRatVal
  - imsa\_UtRatInfo, 102
- UtService
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- UtServiceRat
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- UtServiceStatus
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- UtSvcStatus
  - imsa\_UtSvcStatusInfo, 102
- uusInfo
  - voice\_allCallsUUSInfo, 1112
- VDOP
  - loc\_precisionDilution, 163
- val
  - unpack\_qos\_IPv6TrafCls\_t, 908
  - unpack\_qos\_Tos\_t, 927
- val\_3GPP2Pri
  - unpack\_qos\_swiQosFlow\_t, 926
- val\_3GPPImCn
  - unpack\_qos\_swiQosFlow\_t, 926
- val\_3GPPResResidualBER

- unpack\_qos\_swiQosFlow\_t, 926
- val\_3GPPSigInd
  - unpack\_qos\_swiQosFlow\_t, 926
- val\_3GPPTraHdlPri
  - unpack\_qos\_swiQosFlow\_t, 926
- ValidBitmasks
  - swidms\_supportedIntBitmaskTlv, 617
- validCard
  - uim\_validCardStatus, 653
- validCardLength
  - uim\_validCardStatus, 653
- ValidMask
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- validMask
  - loc\_satelliteInfo, 166
- ValidityCW0
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 882
- ValidityCW1
  - unpack\_nas\_SLQSSwiGetLteCQI\_t, 882
- Value
  - pack\_swiaudio\_SLQSSetM2MSpkrGain\_t, 503
  - unpack\_swiaudio\_SLQSGetM2MSpkrGain\_t, 954
- value
  - pack\_dms\_UIMSetPINProtection\_t, 397
  - pack\_dms\_UIMVerifyPIN\_t, 398
- value\_length
  - DMScustSettingInfo, 75
  - pack\_dms\_SetCustFeaturesV2\_t, 386
- verValid
  - loc\_AppProviderInfoTlv, 150
- verboseSessnEndReason
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 1087
- verboseSessnEndReasonType
  - unpack\_wds\_SLQSSetPacketSrvStatusCallback\_t, 1087
- verifyLeft
  - uim\_personalizationStatus, 638
  - uim\_remainingRetries, 645
- verifyPIN
  - pack\_uim\_VerifyPin\_t, 536
- verifyRetriesLeft
  - unpack\_dms\_UIMGetControlKeyStatus\_t, 731
  - unpack\_dms\_UIMSetControlKeyProtection\_t, 734
  - unpack\_dms\_UIMSetPINProtection\_t, 735
- version
  - loc\_AppProviderInfoTlv, 150
  - unpack\_omaDmFotaTlv\_t, 891
  - unpack\_qos\_swiQosFilter\_t, 922
- versionLen
  - loc\_AppProviderInfoTlv, 150
- versionStr
  - \_litfw\_FirmwareFileInfo, 36
- versionString
  - pack\_dms\_SLQSSwiSetOSInfo\_t, 392
  - unpack\_dms\_SLQSSwiGetOSInfo\_t, 725
- versionlength
  - unpack\_omaDmFotaTlv\_t, 891
- vertConfidence
  - pack\_loc\_SLQSLOCInjectPosition\_t, 418
- vertReliability
  - pack\_loc\_SLQSLOCInjectPosition\_t, 418
- vertUnc
  - pack\_loc\_SLQSLOCInjectPosition\_t, 418
- VerticalUncertainty
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, 905
- VirtStream
  - nas\_protocolSubtypeElement, 290
- voice.h
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATA, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUI-TASYNC, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_DATACIRCUI-TSYNC, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_FAX, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_NONE, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_PACKETACCESS, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_PADACCESS, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_SMS, 1580
  - LITE\_VOICE\_SUPS\_SRV\_CLASS\_VOICE, 1580
- voice.h, 1574
  - liteServiceClassInformation, 1580
  - MAXVOICEUSSDLENGTH, 1579
  - pack\_voice\_AnswerUSSD, 1580
  - pack\_voice\_CancelUSSD, 1581
  - pack\_voice\_OriginateUSSD, 1581
  - pack\_voice\_SLQSOriginateUSSD, 1581
  - pack\_voice\_SLQSVoiceALSSelectLine, 1582
  - pack\_voice\_SLQSVoiceALSSetLineSwitching, 1582
  - pack\_voice\_SLQSVoiceAnswerCall, 1582
  - pack\_voice\_SLQSVoiceBindSubscription, 1583
  - pack\_voice\_SLQSVoiceBurstDTMF, 1583
  - pack\_voice\_SLQSVoiceDialCall, 1584
  - pack\_voice\_SLQSVoiceEndCall, 1584
  - pack\_voice\_SLQSVoiceGetAllCallInfo, 1584
  - pack\_voice\_SLQSVoiceGetCLIP, 1586
  - pack\_voice\_SLQSVoiceGetCLIR, 1587
  - pack\_voice\_SLQSVoiceGetCNAP, 1587
  - pack\_voice\_SLQSVoiceGetCOLP, 1587
  - pack\_voice\_SLQSVoiceGetCOLR, 1588
  - pack\_voice\_SLQSVoiceGetCallBarring, 1585
  - pack\_voice\_SLQSVoiceGetCallForwardingStatus, 1585
  - pack\_voice\_SLQSVoiceGetCallInfo, 1585
  - pack\_voice\_SLQSVoiceGetCallWaiting, 1586
  - pack\_voice\_SLQSVoiceGetConfig, 1588
  - pack\_voice\_SLQSVoiceIndicationRegister, 1588
  - pack\_voice\_SLQSVoiceManageCalls, 1589
  - pack\_voice\_SLQSVoiceOrigUSSDNoWait, 1589
  - pack\_voice\_SLQSVoiceSendFlash, 1589
  - pack\_voice\_SLQSVoiceSetCallBarringPassword, 1590
  - pack\_voice\_SLQSVoiceSetConfig, 1590



- pack\_voice\_SLQSVoiceSetPreferredPrivacy, 1590
- pack\_voice\_SLQSVoiceSetSUPSService, 1591
- pack\_voice\_SLQSVoiceStartContDTMF, 1591
- pack\_voice\_SLQSVoiceStopContDTMF, 1592
- unpack\_voice\_AnswerUSSD, 1592
- unpack\_voice\_AnswerUSSD\_t, 1580
- unpack\_voice\_CancelUSSD, 1593
- unpack\_voice\_CancelUSSD\_t, 1580
- unpack\_voice\_DTMFEventCallback\_ind, 1593
- unpack\_voice\_OTASPStatusCallback\_ind, 1594
- unpack\_voice\_OriginateUSSD, 1593
- unpack\_voice\_OriginateUSSD\_t, 1580
- unpack\_voice\_SLQSOriinateUSSD, 1594
- unpack\_voice\_SLQSVoiceALSSelectLine, 1594
- unpack\_voice\_SLQSVoiceALSSelectLine\_t, 1580
- unpack\_voice\_SLQSVoiceALSSetLineSwitching, 1595
- unpack\_voice\_SLQSVoiceALSSetLineSwitching\_t, 1580
- unpack\_voice\_SLQSVoiceAnswerCall, 1595
- unpack\_voice\_SLQSVoiceBindSubscription, 1596
- unpack\_voice\_SLQSVoiceBindSubscription\_t, 1580
- unpack\_voice\_SLQSVoiceBurstDTMF, 1596
- unpack\_voice\_SLQSVoiceDialCall, 1596
- unpack\_voice\_SLQSVoiceEndCall, 1597
- unpack\_voice\_SLQSVoiceGetAllCallInfo, 1597
- unpack\_voice\_SLQSVoiceGetCLIP, 1599
- unpack\_voice\_SLQSVoiceGetCLIR, 1599
- unpack\_voice\_SLQSVoiceGetCNAP, 1599
- unpack\_voice\_SLQSVoiceGetCOLP, 1600
- unpack\_voice\_SLQSVoiceGetCOLR, 1600
- unpack\_voice\_SLQSVoiceGetCallBarring, 1597
- unpack\_voice\_SLQSVoiceGetCallForwarding-Status, 1598
- unpack\_voice\_SLQSVoiceGetCallInfo, 1598
- unpack\_voice\_SLQSVoiceGetCallWaiting, 1598
- unpack\_voice\_SLQSVoiceGetConfig, 1600
- unpack\_voice\_SLQSVoiceIndicationRegister, 1601
- unpack\_voice\_SLQSVoiceIndicationRegister\_t, 1580
- unpack\_voice\_SLQSVoiceManageCalls, 1601
- unpack\_voice\_SLQSVoiceOrigUSSDNoWait, 1601
- unpack\_voice\_SLQSVoiceOrigUSSDNoWait\_t, 1580
- unpack\_voice\_SLQSVoiceSUPSCallback\_ind, 1604
- unpack\_voice\_SLQSVoiceSendFlash, 1602
- unpack\_voice\_SLQSVoiceSetCallBarringPassword, 1602
- unpack\_voice\_SLQSVoiceSetConfig, 1602
- unpack\_voice\_SLQSVoiceSetPreferredPrivacy, 1603
- unpack\_voice\_SLQSVoiceSetPreferredPrivacy\_t, 1580
- unpack\_voice\_SLQSVoiceSetSUPSService, 1603
- unpack\_voice\_SLQSVoiceStartContDTMF, 1603
- unpack\_voice\_SLQSVoiceStopContDTMF, 1604
- unpack\_voice\_SUPSNotificationCallback\_ind, 1604
- unpack\_voice\_USSDNotificationCallback\_ind, 1605
- unpack\_voice\_VoiceInfoRecCallback\_ind, 1605
- unpack\_voice\_allCallStatusCallback\_ind, 1592
- unpack\_voice\_voicePrivacyChangeCallback\_ind, 1605
- voice\_CLIPResp, 1132
  - ActiveStatus, 1132
  - ProvisionStatus, 1132
- voice\_CLIRResp, 1132
  - ActiveStatus, 1133
  - ProvisionStatus, 1133
- voice\_CNAPResp, 1133
  - ActiveStatus, 1134
  - ProvisionStatus, 1134
- voice\_COLPResp, 1134
  - ActiveStatus, 1134
  - ProvisionStatus, 1134
- voice\_COLRResp, 1134
  - ActiveStatus, 1135
  - ProvisionStatus, 1135
- voice\_CUGInfo, 1137
  - CUGIndex, 1137
  - SuppOA, 1137
  - SuppPrefCUG, 1137
- voice\_DTMFInfo, 1139
  - callID, 1139
  - DTMFEvent, 1139
  - digitBuff, 1139
  - digitCnt, 1139
- voice\_DTMFLengths, 1139
  - DTMFInterdigitInterval, 1140
  - DTMFPulseWidth, 1140
- voice\_ECTNum, 1140
  - ECTCallState, 1141
  - number, 1141
  - presentationInd, 1141
- voice\_NSSAudioCtrl, 1146
  - downLink, 1146
  - upLink, 1146
- voice\_SUPSInfo, 1155
  - isModByCC, 1155
  - svcType, 1155
- voice\_USSDNotificationNetworkInfo, 1155
  - networkInfo, 1156
  - tlvPresent, 1156
- voice\_USSInfo, 1156
  - ussDCS, 1156
  - ussData, 1156
  - ussLen, 1156
- voice\_UUSInfo, 1156
  - UUSData, 1157
  - UUSDataLen, 1157
  - UUSDCs, 1157
  - UUSType, 1157
- voice\_airTimer, 1110

- airTimerValue, 1110
- namID, 1110
- voice\_allCallsAlphaIDInfo, 1110
  - AlphaIDInfo, 1111
  - callID, 1111
- voice\_allCallsDiagInfo, 1111
  - callID, 1111
  - DiagInfo, 1111
- voice\_allCallsUUSInfo, 1111
  - callID, 1112
  - uusInfo, 1112
- voice\_alphaIDInfo, 1112
  - alphaDcs, 1112
  - alphaLen, 1112
  - alphaText, 1112
- voice\_arrAlertingPattern, 1112
  - alertingPattern, 1113
  - callID, 1113
  - numInstances, 1113
- voice\_arrAlertingType, 1113
  - AlertingType, 1114
  - callID, 1114
  - numInstances, 1114
- voice\_arrAlphaID, 1114
  - allCallsAlphaIDInfoArr, 1114
  - numInstances, 1114
- voice\_arrCallEndReason, 1115
  - callEndReason, 1116
  - callID, 1116
  - numInstances, 1116
- voice\_arrCallInfo, 1116
  - getAllCallInfo, 1116
  - numInstances, 1116
- voice\_arrCalledPartyNum, 1114
  - CalledPartyNum, 1115
  - numInstances, 1115
- voice\_arrConnectPartyNum, 1116
  - ConnectedPartyNum, 1117
  - numInstances, 1117
- voice\_arrDiagInfo, 1117
  - DiagInfo, 1117
  - numInstances, 1117
- voice\_arrRedirPartyNum, 1117
  - numInstances, 1118
  - RedirPartyNum, 1118
- voice\_arrRemotePartyName, 1118
  - GetAllCallRmtPtyName, 1118
  - numInstances, 1118
- voice\_arrRemotePartyNum, 1118
  - numInstances, 1119
  - RmtPtyNum, 1119
- voice\_arrSvcOption, 1119
  - callID, 1119
  - numInstances, 1119
  - srvOption, 1119
- voice\_arrUUSInfo, 1120
  - AllCallsUUSInfo, 1120
  - numInstances, 1120
- voice\_burstDTMFInfo, 1120
  - digitCnt, 1121
  - pCallID, 1121
  - pDigitBuff, 1121
- voice\_callFWExtInfo, 1125
  - noReplyTimer, 1126
  - numLen, 1126
  - numPlan, 1127
  - numType, 1127
  - number, 1126
  - PI, 1127
  - SI, 1127
  - SvcClass, 1127
  - SvcStatus, 1127
- voice\_callFWInfo, 1127
  - noReplyTimer, 1127
  - numLen, 1128
  - number, 1127
  - SvcClass, 1128
  - SvcStatus, 1128
- voice\_callFwdTypeAndPlan, 1124
  - numberPlan, 1125
  - numberType, 1125
- voice\_callInfo, 1128
  - callID, 1129
  - callState, 1129
  - callType, 1129
  - direction, 1129
  - mode, 1129
- voice\_calledPartyInfo, 1121
  - numLen, 1122
  - numPlan, 1122
  - numType, 1122
  - number, 1122
  - PI, 1122
  - SI, 1122
- voice\_calledPartySubAdd, 1122
  - extBit, 1123
  - oddEvenInd, 1123
  - subAddr, 1123
  - subAddrLen, 1123
  - subAddrType, 1123
- voice\_callerIDInfo, 1123
  - callerID, 1124
  - callerIDLen, 1124
  - PI, 1124
- voice\_callingPartyInfo, 1129
  - numLen, 1131
  - numPlan, 1131
  - numType, 1131
  - number, 1131
  - PI, 1131
  - SI, 1131
- voice\_ccSUPSType, 1131
  - reason, 1131
  - svcType, 1132
- voice\_connectNumInfo, 1135
  - callerID, 1136

- callerIDLen, 1136
- numPlan, 1136
- numPresInd, 1136
- numType, 1137
- screeningInd, 1137
- voice\_curAMRConfig, 1137
  - gsmAmrStat, 1138
  - wcdmaAmrStat, 1138
- voice\_diagInfo, 1138
  - diagInfoLen, 1138
  - diagnosticInfo, 1138
- voice\_extDispRecInfo, 1141
  - dispType, 1141
  - extDispInfo, 1141
  - extDispInfoLen, 1141
- voice\_getAllCallInformation, 1142
  - ALS, 1142
  - Callinfo, 1142
  - isEmpty, 1142
- voice\_getAllCallRmtPtyName, 1142
  - callID, 1143
  - RemotePartyName, 1143
- voice\_getAllCallRmtPtyNum, 1143
  - callID, 1143
  - RemotePartyNum, 1143
- voice\_getCallFWExtInfo, 1143
  - CallFWExtInfo, 1144
  - numInstances, 1144
- voice\_getCallFWInfo, 1144
  - CallFWInfo, 1144
  - numInstances, 1144
- voice\_lineCtrlInfo, 1144
  - polarityIncluded, 1145
  - pwrDenialTime, 1145
  - revPolarity, 1145
  - toggleMode, 1145
- voice\_newPwdData, 1145
  - newPwd, 1146
  - newPwdAgain, 1146
- voice\_peerNumberInfo, 1146
  - callID, 1147
  - numLen, 1148
  - numPI, 1148
  - numPlan, 1148
  - numSI, 1148
  - numType, 1148
  - number, 1148
- voice\_prefVoiceSO, 1148
  - evrcCapability, 1150
  - homeOrigVoiceSO, 1150
  - homePageVoiceSO, 1150
  - namID, 1150
  - roamOrigVoiceSO, 1150
- voice\_redirNumInfo, 1150
  - numLen, 1151
  - numPlan, 1151
  - numType, 1151
  - number, 1151
  - PI, 1151
  - reason, 1151
  - SI, 1151
- voice\_remotePartyName, 1152
  - callerName, 1152
  - codingScheme, 1152
  - nameLen, 1152
  - namePI, 1152
- voice\_remotePartyNum, 1152
  - numLen, 1153
  - presentationInd, 1153
  - remPartyNumber, 1153
- voice\_roamTimer, 1153
  - namID, 1154
  - roamTimerValue, 1154
- voice\_signalInfo, 1154
  - alertPitch, 1154
  - signal, 1154
  - signalType, 1154
- VoiceDomainPref
  - nas\_VoiceDomainPrefTlv, 344
- VoiceNumber
  - unpack\_dms\_GetVoiceNumber\_t, 693
- voiceNumberSize
  - unpack\_dms\_GetVoiceNumber\_t, 693
- voicePrivacy
  - unpack\_voice\_voicePrivacyChangeCallback\_ind\_t, 1048
- voiceSvc
  - pack\_voice\_SLQSVoiceSetSUPSService\_t, 554
- VoipRat
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- VoipRatVal
  - imsa\_VoipRatInfo, 103
- VoipService
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- VoipServiceRat
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VoipServiceStatus
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VoipSvcStatus
  - imsa\_VoipSvcStatusInfo, 103
- VolValue
  - pack\_audio\_SLQSSetAudioVolTLBConfig\_t, 378
- VoltStat
  - dms\_VoltageTlv, 74
- VoltTlv
  - unpack\_dms\_SwiEventReportCallBack\_ind\_t, 729
- Voltage
  - dms\_VoltageTlv, 74
- Volume
  - pack\_audio\_SLQSGetAudioVolTLBConfig\_t, 374
  - pack\_audio\_SLQSSetAudioProfile\_t, 377
  - pack\_audio\_SLQSSetAudioVolTLBConfig\_t, 378
  - unpack\_audio\_SLQSGetAudioProfile\_t, 658

- unpack\_swiaudio\_SLQSGetM2MAudioProfile\_t, 952
- voteForInit
  - uim\_registerRefresh, 644
- VsServiceRat
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VsServiceStatus
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VtRat
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- VtRatVal
  - imsa\_VtRatInfo, 104
- VtService
  - unpack\_imsa\_SLQSImsaSvcStatusCallBack\_ind\_t, 760
- VtServiceRat
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VtServiceStatus
  - unpack\_imsa\_SLQSGetIMSAServiceStatus\_t, 756
- VtSvcStatus
  - imsa\_VtSvcStatusInfo, 104
- wAgc
  - nas\_WCDMACellInfoExt, 346
- wAlertMsgLength
  - UnpackSwiAvmsEventReportConfig, 1103
- WCDMACellInfo
  - nas\_lteWcdmaCellInfo, 254
- WCDMAECIOThreshListLen
  - nas\_WCDMAECIOThresh, 347
- WCDMARSSIThreshListLen
  - nas\_WCDMARSSIThresh, 348
- WCDMASSInfo
  - unpack\_nas\_SLQSNasGetSigInfo\_t, 865
- WDS\_PROFILE\_3GPP
  - wds.h, 1615
- WDS\_PROFILE\_3GPP2
  - wds.h, 1615
- wDescriptionLength
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- wDIBler
  - nas\_WCDMACellInfoExt, 346
- wErrorCode
  - UnpackSwiAvmsEventReportDataSessionStatus, 1104
- wHTTPStatus
  - UnpackSwiAvmsEventReportHTTPStatus, 1105
- wNameLength
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- WORD
  - SwiDataTypes.h, 1517
- wSessionStatus
  - UnpackSwiAvmsEventReportNotification, 1107
- wTxAgc
  - nas\_WCDMACellInfoExt, 346
- wUpdateCompeteStatus
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- wUserInputTimeout
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- UnpackSwiAvmsEventReportConfig, 1103
- UnpackSwiAvmsEventReportConnectionRequest, 1103
- wVersionLength
  - UnpackSwiAvmsEventReportBinaryUpdate-SessionInfo, 1102
- wWamsChangeMask
  - UnpackSwiAvmsEventReportWAMSParm-Change, 1109
- wake\_mask
  - pack\_rms\_SetSMSWake\_t, 487
  - unpack\_rms\_GetSMSWake\_t, 929
- wcdmaAmrStat
  - voice\_curAMRConfig, 1138
- wcdmaRRCTest
  - nas\_WCDMAInfoLTENeighborCell, 347
- Wdisable
  - unpack\_dms\_SLQSDmsSwiGetPCInfo\_t, 703
- wds.h
  - QMI\_LITE\_WDS\_CURRENT\_CALL\_DB\_MASK, 1616
  - QMI\_LITE\_WDS\_LAST\_CALL\_DB\_MASK, 1616
- wds.h, 1606
  - BYT\_STAT\_STAT\_MASK, 1614
  - liteQmiDataBearerMasks, 1616
  - PACK\_WDS\_IPV4, 1615
  - PACK\_WDS\_IPV6, 1615
  - pack\_wds\_DHCPv4ClientLeaseChange, 1616
  - pack\_wds\_GetAutoconnect, 1617
  - pack\_wds\_GetByteTotals, 1617
  - pack\_wds\_GetConnectionRate, 1617
  - pack\_wds\_GetDataBearerTechnology, 1618
  - pack\_wds\_GetDefaultProfile, 1618
  - pack\_wds\_GetDefaultProfileNum, 1619
  - pack\_wds\_GetDefaultProfileV2, 1619
  - pack\_wds\_GetDormancyState, 1619
  - pack\_wds\_GetLastMobileIPError, 1620
  - pack\_wds\_GetMobileIP, 1620
  - pack\_wds\_GetMobileIPProfile, 1621
  - pack\_wds\_GetPacketStatistics, 1621
  - pack\_wds\_GetPacketStatus, 1622
  - pack\_wds\_GetSessionDuration, 1622
  - pack\_wds\_GetSessionDurationV2, 1622
  - pack\_wds\_GetSessionState, 1623
  - pack\_wds\_RMSetTransferStatistics, 1623
  - pack\_wds\_SLQSCreateProfile, 1627
  - pack\_wds\_SLQSDestroyProfile, 1627
  - pack\_wds\_SLQSGet3GPPConfigItem, 1627
  - pack\_wds\_SLQSGetCurrDataSystemStat, 1628
  - pack\_wds\_SLQSGetCurrentChannelRate, 1628
  - pack\_wds\_SLQSGetDUNCallInfo, 1629
  - pack\_wds\_SLQSGetDataBearerTechnology, 1629
  - pack\_wds\_SLQSGetProfileSettings, 1630

- pack\_wds\_SLQSGetProfileSettingsV2, 1630
- pack\_wds\_SLQSGetRuntimeSettings, 1630
- pack\_wds\_SLQSModifyProfile, 1631
- pack\_wds\_SLQSResetPacketStatics, 1631
- pack\_wds\_SLQSSetDHCPv4ClientConfig, 1633
- pack\_wds\_SLQSSetLoopback, 1633
- pack\_wds\_SLQSSetDHCPv4ClientConfig, 1634
- pack\_wds\_SLQSSetLoopback, 1634
- pack\_wds\_SLQSSet3GPPConfigItem, 1632
- pack\_wds\_SLQSSetIPFamilyPreference, 1632
- pack\_wds\_SLQSSetWdsEventCallback, 1632
- pack\_wds\_SLQSStartDataSession, 1634
- pack\_wds\_SLQSStopDataSession, 1635
- pack\_wds\_SLQSSwiProfileChangeCallback, 1635
- pack\_wds\_SLQSWdsGoActive, 1636
- pack\_wds\_SLQSWdsGoDormant, 1636
- pack\_wds\_SLQSWdsSetEventReport, 1637
- pack\_wds\_SLQSWdsSwiPDPRuntimeSettings, 1637
- pack\_wds\_SetAutoconnect, 1624
- pack\_wds\_SetDefaultProfile, 1624
- pack\_wds\_SetDefaultProfileNum, 1624
- pack\_wds\_SetMobileIP, 1625
- pack\_wds\_SetMobileIPParameters, 1625
- pack\_wds\_SetMobileIPProfile, 1626
- pack\_wds\_SetMuxID, 1626
- unpack\_wds\_DHCPv4ClientLease\_ind, 1637
- unpack\_wds\_DHCPv4ClientLeaseChange, 1638
- unpack\_wds\_DHCPv4ClientLeaseChange\_t, 1615
- unpack\_wds\_GetAutoconnect, 1638
- unpack\_wds\_GetByteTotals, 1639
- unpack\_wds\_GetConnectionRate, 1639
- unpack\_wds\_GetDataBearerTechnology, 1639
- unpack\_wds\_GetDefaultProfile, 1640
- unpack\_wds\_GetDefaultProfileNum, 1640
- unpack\_wds\_GetDefaultProfileV2, 1640
- unpack\_wds\_GetDormancyState, 1641
- unpack\_wds\_GetLastMobileIPError, 1641
- unpack\_wds\_GetMobileIP, 1641
- unpack\_wds\_GetMobileIPProfile, 1642
- unpack\_wds\_GetPacketStatistics, 1642
- unpack\_wds\_GetPacketStatus, 1642
- unpack\_wds\_GetSessionDuration, 1643
- unpack\_wds\_GetSessionDurationV2, 1643
- unpack\_wds\_GetSessionState, 1643
- unpack\_wds\_RMSetTransferStatistics, 1644
- unpack\_wds\_RMTransferStatistics\_ind, 1644
- unpack\_wds\_RMTransferStatistics\_ind\_t, 1615
- unpack\_wds\_SLQSCreateProfile, 1647
- unpack\_wds\_SLQSDUNCallInfoCallBack\_ind, 1647
- unpack\_wds\_SLQSDeleteProfile, 1647
- unpack\_wds\_SLQSGet3GPPConfigItem, 1648
- unpack\_wds\_SLQSGetCurrDataSystemStat, 1648
- unpack\_wds\_SLQSGetCurrentChannelRate, 1648
- unpack\_wds\_SLQSGetDUNCallInfo, 1649
- unpack\_wds\_SLQSGetDataBearerTechnology, 1649
- unpack\_wds\_SLQSGetProfileSettings, 1649
- unpack\_wds\_SLQSGetProfileSettingsV2, 1650
- unpack\_wds\_SLQSGetRuntimeSettings, 1650
- unpack\_wds\_SLQSModifyProfile, 1650
- unpack\_wds\_SLQSResetPacketStatics, 1651
- unpack\_wds\_SLQSResetPacketStatics\_t, 1615
- unpack\_wds\_SLQSSetDHCPv4ClientConfig, 1653
- unpack\_wds\_SLQSSetLoopback, 1653
- unpack\_wds\_SLQSSetDHCPv4ClientConfig, 1653
- unpack\_wds\_SLQSSetDHCPv4ClientConfig\_t, 1615
- unpack\_wds\_SLQSSetLoopback, 1654
- unpack\_wds\_SLQSSetLoopback\_t, 1615
- unpack\_wds\_SLQSSet3GPPConfigItem, 1651
- unpack\_wds\_SLQSSet3GPPConfigItem\_t, 1615
- unpack\_wds\_SLQSSetIPFamilyPreference, 1651
- unpack\_wds\_SLQSSetPacketSrvStatusCallback, 1652
- unpack\_wds\_SLQSSetWdsEventCallback, 1652
- unpack\_wds\_SLQSSetWdsEventCallback\_ind, 1652
- unpack\_wds\_SLQSSetWdsEventCallback\_t, 1615
- unpack\_wds\_SLQSStartDataSession, 1654
- unpack\_wds\_SLQSStopDataSession, 1654
- unpack\_wds\_SLQSStopDataSession\_t, 1615
- unpack\_wds\_SLQSSwiProfileChangeCallback, 1655
- unpack\_wds\_SLQSSwiProfileChangeCallback\_Ind, 1655
- unpack\_wds\_SLQSSwiProfileChangeCallback\_t, 1615
- unpack\_wds\_SLQSWdsGoActive, 1655
- unpack\_wds\_SLQSWdsGoActive\_t, 1615
- unpack\_wds\_SLQSWdsGoDormant, 1656
- unpack\_wds\_SLQSWdsGoDormant\_t, 1615
- unpack\_wds\_SLQSWdsSetEventReport, 1656
- unpack\_wds\_SLQSWdsSetEventReport\_t, 1615
- unpack\_wds\_SLQSWdsSwiPDPRuntimeSettings, 1656
- unpack\_wds\_SetAutoconnect, 1644
- unpack\_wds\_SetAutoconnect\_t, 1615
- unpack\_wds\_SetDefaultProfile, 1645
- unpack\_wds\_SetDefaultProfile\_t, 1615
- unpack\_wds\_SetDefaultProfileNum, 1645
- unpack\_wds\_SetDefaultProfileNum\_t, 1615
- unpack\_wds\_SetMobileIP, 1645
- unpack\_wds\_SetMobileIP\_t, 1615
- unpack\_wds\_SetMobileIPParameters, 1646
- unpack\_wds\_SetMobileIPParameters\_t, 1615
- unpack\_wds\_SetMobileIPProfile, 1646
- unpack\_wds\_SetMuxID, 1646
- unpack\_wds\_SetMuxID\_t, 1615
- UnpackQmiProfileInfo, 1615
- UnpackQmiProfileInfoV2, 1616
- WDS\_PROFILE\_3GPP, 1615
- WDS\_PROFILE\_3GPP2, 1615

- wds\_ChannelRateTlv, 1158
  - ChannelRate, 1158
  - TlvPresent, 1158
- wds\_ConnStatusTlv, 1159
  - MDMConnStatus, 1159
  - TlvPresent, 1159
- wds\_DHCPLeaseOptTlv, 1162
  - numOpt, 1162
  - optList, 1162
  - optListData, 1163
  - TlvPresent, 1163
- wds\_DHCPLeaseStateTlv, 1163
  - leaseState, 1163
  - TlvPresent, 1163
- wds\_DHCPOpt, 1163
  - optCode, 1164
  - optValLen, 1164
  - pOptVal, 1164
- wds\_DHCPProfileIdTlv, 1164
  - profileId, 1164
  - profileType, 1164
  - TlvPresent, 1164
- wds\_DHCPv4HWConfig, 1164
  - chaddr, 1165
  - chaddrLen, 1165
  - hwType, 1165
- wds\_DHCPv4Option, 1165
  - optCode, 1166
  - optVal, 1166
  - optValLen, 1166
- wds\_DHCPv4OptionList, 1166
  - numOpt, 1166
  - pOptList, 1166
- wds\_DHCPv4ProfileId, 1166
  - profileId, 1167
  - profileType, 1167
- wds\_DataBearTechTlv, 1160
  - DataBearerTech, 1161
  - TlvPresent, 1161
- wds\_DataULongLongTlv, 1161
  - TlvPresent, 1162
  - ulldata, 1162
- wds\_DataULongTlv, 1162
  - TlvPresent, 1162
  - ulData, 1162
- wds\_Domain, 1167
  - domainLen, 1167
  - domainName, 1167
- wds\_DomainNameList, 1167
  - domain, 1168
  - numInstances, 1168
- wds\_DormStatTlv, 1168
  - DormancyStat, 1168
  - TlvPresent, 1168
- wds\_GPRSQoS, 1168
  - delayClass, 1169
  - meanThroughputClass, 1169
  - peakThroughputClass, 1169
  - precedenceClass, 1169
  - reliabilityClass, 1169
- wds\_IPV6AddressInfo, 1170
  - IPAddressV6, 1170
  - IPv6PrefixLen, 1170
- wds\_IPV6GWAddressInfo, 1170
  - gwAddressV6, 1171
  - gwV6PrefixLen, 1171
- wds\_IPv4AdTlv, 1169
  - IPv4Addr, 1170
  - TlvPresent, 1170
- wds\_LastMdmCallEndRsnTlv, 1171
  - CallEndReason, 1171
  - TlvPresent, 1171
- wds\_PCSCFFQDNAddress, 1171
  - fqdnAddr, 1172
  - fqdnLen, 1172
- wds\_PCSCFFQDNAddressList, 1172
  - numInstances, 1172
  - pcsfFQDNAddress, 1172
- wds\_PCSCFIPv4ServerAddressList, 1172
  - numInstances, 1173
  - pcscfIPv4Addr, 1173
- wds\_ProfileIdentifier, 1173
  - profileIndex, 1174
  - profileType, 1174
- wds\_RXBytesOKTlv, 1175
  - RxByteOKCnt, 1175
  - TlvPresent, 1175
- wds\_TXBytesOKTlv, 1177
  - TlvPresent, 1177
  - TxByteOKCnt, 1177
- wds\_TrStatInd, 1176
  - statsMask, 1177
  - statsPeriod, 1177
- wds\_UMTSMinQoS, 1177
  - deliveryErrSDU, 1179
  - grntDownlinkBitrate, 1179
  - grntUplinkBitrate, 1179
  - maxDownlinkBitrate, 1179
  - maxSDUSize, 1179
  - maxUplinkBitrate, 1179
  - qosDeliveryOrder, 1179
  - resBerRatio, 1179
  - sduErrorRatio, 1179
  - trafficClass, 1179
  - trafficPriority, 1179
  - transferDelay, 1179
- wds\_channelRate, 1158
  - CurrChanRxRate, 1158
  - CurrChanTxRate, 1158
- wds\_currNetworkInfo, 1159
  - NetworkType, 1160
  - RATMask, 1160
  - SOMask, 1160
- wds\_profileChange, 1173
  - profileIdx, 1173
  - profileType, 1173



- wds\_profileInfo, [1174](#)
  - SlqsProfile3GPP, [1174](#)
  - SlqsProfile3GPP2, [1174](#)
- wds\_sourceOfChange, [1175](#)
  - source, [1175](#)
- wds\_transferStatInd, [1175](#)
  - StatsMask, [1176](#)
  - StatsPeriod, [1176](#)
- wdsDhcpv4HwConfig, [1179](#)
  - chaddr, [1180](#)
  - chaddrLen, [1180](#)
  - hwType, [1180](#)
- wdsDhcpv4Option, [1180](#)
  - optCode, [1180](#)
  - optVal, [1181](#)
  - optValLen, [1181](#)
- wdsDhcpv4OptionList, [1181](#)
  - numOpt, [1181](#)
  - pOptList, [1181](#)
- wdsDhcpv4ProfileId, [1181](#)
  - profileId, [1182](#)
  - profileType, [1182](#)
- word
  - swi\_uint256\_t, [613](#)
- writeRecord
  - pack\_uim\_SLQSUIWriteRecord\_t, [533](#)
- writeTransparent
  - pack\_uim\_SLQSUIWriteTransparent\_t, [534](#)
- xAxis
  - sensorData\_t, [592](#)
- xferStatAvail
  - unpack\_wds\_SLQSSetWdsEventCallback\_ind\_t, [1093](#)
- xid
  - pack\_loc\_SLQSLOCGetBestAvailPos\_t, [412](#)
  - pack\_qmi\_t, [485](#)
  - unpack\_qmi\_t, [906](#)
- xtra\_start\_gps\_minutes
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [905](#)
- xtra\_start\_gps\_week
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [906](#)
- xtra\_valid\_duration\_hours
  - unpack\_pds\_SLQSGetGPSSStateInfo\_t, [906](#)
- yAxis
  - sensorData\_t, [592](#)
- year
  - nas\_timeInfo, [334](#)
  - nas\_UniversalTime, [343](#)
- zAxis
  - sensorData\_t, [592](#)